Computing and Informatics, Vol. 42, 2023, 525–545, doi: 10.31577/cai_2023_3_525

LARGE SCALE FINE-TUNED TRANSFORMERS MODELS APPLICATION FOR BUSINESS NAMES GENERATION

Mantas Lukauskas

Department of Applied Mathematics
Kaunas University of Technology
K. Donelaičio st. 73, LT-44249 Kaunas, Lithuania
e-mail: mantas.lukauskas@ktu.lt

Tomas Rasymas

Hostinger, UAB
Jonavos st. 60C, LT-44192 Kaunas, Lithuania
e-mail: tomas.rasymas@hostinger.com

Matas Minelga, Domas Vaitmonas

Zyro Inc, UAB
Jonavos st. 60C, LT-44192 Kaunas, Lithuania
e-mail: {matas, domas}@zyro.com

Abstract. Natural language processing (NLP) involves the computer analysis and processing of human languages using a variety of techniques aimed at adapting various tasks or computer programs to linguistically process natural language. Currently, NLP is increasingly applied to a wide range of real-world problems. These tasks can vary from extracting meaningful information from unstructured data, analyzing sentiment, translating text between languages, to generating human-level text autonomously. The goal of this study is to employ transformer-based natural language models to generate high-quality business names. Specifically, this work investigates whether larger models, which require more training time, yield better results for generating relatively short texts, such as business names. To achieve

this, we utilize different transformer architectures, including both freely available and proprietary models, and compare their performance. Our dataset comprises 250 928 observations of business names. Based on the perplexity metric, the top-performing model in our study is the GPT2-Medium model. However, our findings reveal a discrepancy between human evaluation and perplexity-based assessment. According to human evaluation, the best results are obtained using the GPT-Neo-1.3B model. Interestingly, the larger GPT-Neo-2.7B model yields poorer results, with its performance not being statistically different from that of the GPT-Neo-125M model, which is 20 times smaller.

Keywords: Natural language processing, NLP, natural language generation, NLG, transformers

Mathematics Subject Classification 2010: 68-T50

1 INTRODUCTION

Every year, the amount of data is increasing at an extremely fast rate, leading to an ever wider application of artificial intelligence. The continuous improvement of artificial intelligence and machine learning is leading to an increasing search for the wider application of these technological solutions not only to structured data but also to unstructured data. One of the areas of unstructured data analysis where artificial intelligence can be applied particularly widely is natural language processing (also known as NLP). Natural language processing is the computer analysis and processing of natural language (which can be delivered as well as written) using a variety of technologies aimed at linguistically adapted various tasks or computer programs in human languages. While this seems like a whole new area that is increasingly being talked about, the development of this area wants to be achieved at the turn of the century. At the beginning of the 20$^{\text{th}}$ century, Andrey Andreyevich Markov introduced a theory of random stochastic processes/circuits. All of this is now known as Markov chains or Markov processes. 1902 Markov provided information that these circuits can predict the next element of the circuit using only the last element. All this was adapted to a data set larger than 20 000 letters, indicating/predicting the future letters of the chain. It must be remembered that computers did not have information at the time, but this theory was still proven at the time. As early as 1954, Georgetown-IBM computers translated Russian sentences into English, using only rule systems. Rule-based systems are sometimes used even now, but the creation of a large number of rules is particularly difficult in the development of various natural language processing models. One of the most widely known neural networks for modeling sequences is recurrent neural networks. Recurrent neural networks were based on David Rumelhart's work in 1986. Hopfield networks – a special kind of RNN – were rediscovered by John Hopfield in

1982. In 1993, a neural history compressor system solved a "Very Deep Learning" task that required more than 1 000 subsequent layers in an RNN that unfolded in time. Recursive neural networks (RNNs) have been developed to better track prediction results and get the work needed to do so. However, these neural networks had a number of problems, as they encountered the varnishing gradient problem when they could not capture longer sequences. For this reason, recurrent neural networks evolved into LSTMs. Long-short term memory neural networks (LSTM) are a type of recurrent neural networks that allow capturing not only past data when the gap between input information and output is small, but also when this gap is much higher. The purpose of using these neural networks is to preserve or in other words remember the values of previous states. This type of neural network was first introduced in 1997 by Hochreiter and Schmidhuber. Recently, due to the possibility of capturing information from the long past, these neural networks are especially often used in practice. These neural networks also have a "circuit" type structure, but their recurrent unit has a completely different structure than simple recurrent neural networks. Another modification of recurrent neural networks quite different from LSTM modification is gated recurrent unit (GRU) networks. This network is similar to an LSTM-type network in that, like LSTMs, various logical elements are used to control the presentation of information. One of the main differences between LSTMs and GRUs is that GRUs do not have memory cells. Of course, the account neural network (CNN) created by Yann Le Cuno in 1980 should not be forgotten either. These neural networks are currently widely used for image processing, but can also be used to process text. Following these discoveries, it seems impossible to achieve more, but in 2017. December. Vaswani et al. published an article "Attention is all you need," which describes the original Transformers model. And currently, most natural language processing tasks are solved using these structure models, in particular. At present, natural language processing is finding more and more different ways to adapt to real practical problems. These tasks can range from finding meaningful information in unstructured data [1], analyzing sentiments [2, 3, 4], and translating text into another language [5, 6] to fully automated human-level text creation [7, 8]. Given that artificial intelligence and natural language processing find so many different uses, this paper examines one of these uses. One application is the creation of human-level texts – in this case, the creation of business names, which are further used in the practical activities of the company. The aim of this study is to apply natural language modeling models of transformer architecture to generate high quality business names. This work aims to determine whether larger and much more training time-requiring models have better results for generating relatively short texts – business names. In doing so, different transformer structure models are used, as well as not only freely available models, but also paid models, which are also included in the comparison. This comparison may allow other authors to more easily select the models used in practice and thus save model training time.

2 TRANSFORMERS STRUCTURE AND MODELS

As mentioned earlier, the transformer architecture is currently the most commonly used in natural language processing. Transformers are deep learning models that are based on a self-awareness mechanism, allowing them to evaluate the significance of each part of the input data differently for the predicted value. Although primarily used in natural language processing, this architecture also finds applications in solving computer vision tasks. Like recurrent neural networks (RNNs), transformers are designed to process sequential input data such as natural language and perform tasks such as natural language translation and text summarization. However, unlike RNNs, transformers process the entire input simultaneously. The attention mechanism provides context for any location in the input sequence. An example of this mechanism in action can be a natural language sentence, where the transformer does not have to process one word at a time but can process the entire sentence or text. For this reason, transformers can perform many more actions in parallel compared to RNN models, significantly reducing the training time of the models.

In 2017, the Google Brain team introduced transformers, which are increasingly chosen for NLP problems, replacing RNN models such as long-short-term memory (LSTM). Additional training parallelization allows training on larger datasets, leading to pre-trained systems such as BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer), which have been trained on large language datasets such as the Wikipedia Corpus and Common Crawl. These systems can be fine-tuned for specific tasks. The structure of the transformer model is presented in Figure 1, where multi-head attention is defined as multihead self-attention and is calculated according to the following formulas [9]:

$$MultiHead(Q,K,V) = Concat\left(head_1,\ldots,head_h\right)W^0, \tag{1}$$

$$head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right), \tag{2}$$

$$Attention(Q,K,V) = softmax\left(QK^T/\sqrt{d_k}V\right), \tag{3}$$

where Q, K and V are the query, keys and values that are used to calculate the focus mechanism. $Concat()$ means connection. h is the number of head. Meanwhile, $W_i^0 \in R^{d_{model}xd_{model}}$ is a matrix of weights of the i^{th} head, W_i^0, W_i^K and W_i^V have the same dimensions $d_{model}xd_k$, where d_{model} is the size of the input embeddings and $d_k = d_{model}/h$.

The attention ($\bullet$) is called the scaled dot-product attention because their weight values are based on the key and queries dot-product. The difference between multi-head focus and masked multi-head focus is that the former allows the model to see the future context and the latter does not, so they are used in encoder and decoder structures, respectively. The feed forward component converts the output from the

last transformer decoder block to a probabilistic distribution using FC layers with a softmax activation function. Each input insertion is accompanied by a position coding to include the order of the input sequence. At the end of each encoder and decoder layer, there is a fully connected feed-forward neural network that processes each input position separately and independently. The network consists of two fully connected layers, between which there is a ReLU activation function. The input and output dimensions of the entire fully connected forward propagation neural network are d_{model}, but the output of the first fully connected layer is d_{ff}, which is typically at least several times larger than d_{model}.

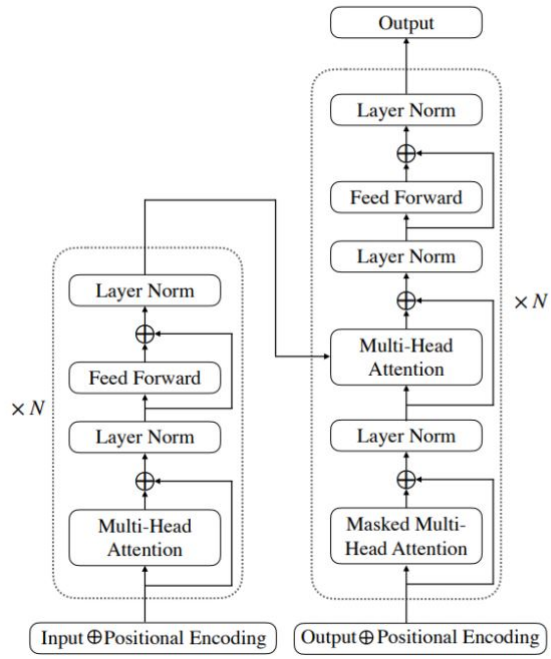


Figure 1. The transformer – model architecture ([9])

Based on the architecture of transformers, many different models have been developed, which often differ in their activation functions, number of layers, and other parameters.

2.1 GPT Models

Just over four years ago, on June 11, 2018, OpenAI released its first Generative Pre-trained Transformer (GPT) model. This initial model was capable of generating relatively long texts. Less than two years later, in February 2019, a significantly larger GPT-2 model was released, followed by the final version of GPT-2 in Novem-

ber 2019. The GPT-2 model, with 1.5 billion parameters, was trained using website texts [10]. It was 10 times larger than its predecessor in terms of both the number of parameters and the training data used.

The most recent OpenAI model is GPT-3 [11], an autoregressive language model trained with an astounding 175 billion parameters to generate highly realistic text. Even without further training, GPT-3 demonstrates remarkable accuracy in a variety of natural language generation tasks. This model challenges the typical supervised learning approach, which involves using specific datasets for specific tasks. GPT-3's performance indicates that language models can learn tasks without explicit supervision, showcasing the potential for more generalized language understanding and applications [12].

2.2 BERT Models

BERT (Bidirectional Encoder Representations from Transformers) models can be described as a pre-training technique developed based on work in contextual representations [13, 13]. The main feature that distinguishes BERT models is their deep bidirectional nature, which is based on unsupervised language representation [13]. DistilBERT is a smaller, more efficient variant of these models, particularly well-suited for solving common language tasks [14]. This model incorporates distillation, where the large-scale model is compressed into a much smaller model. As a result, DistilBERT is about 40 % smaller and 60 % faster while maintaining up to 97 % model accuracy. Several other technical improvements to BERT models exist, such as ALBERT [15], BART [16], DocBERT [17], and Facebook's RoBERTa [18]. However, these models were not deemed suitable for solving the problem at hand and were thus not used in the study.

XLNet builds on the foundations of BERT and GPT models and aims to address their shortcomings. The basic architecture of XLNet is based on the Transformer-XL model [19]. XLNet can learn bidirectional context by maximizing the probability and uses autoregressive formulation because it is based on the Transformer-XL architecture. This avoids the limitations of the BERT model [20]. The Transformer-XL architecture introduces a recurrence mechanism at the segment level into the transformer architecture. This is achieved by saving the hidden states generated from the previous segment and then using them as keys and values when processing the next segment. The permutation language modeling method, like traditional language models, provides one token at a time, depending on the previous context. However, it presents tokens in a random rather than sequential order [21].

3 MATERIALS AND METHODS

This section describes the materials and methods used in this work to perform the above evaluation. We first describe the datasets used, the methods used, and the experimental setup.

3.1 Data

The dataset for this study comprises 350 928 observations, or business names, with 299,964 observations in the training sample and 50,964 observations in the test sample. The process of data collection is illustrated in Figure 2. This data was collected using websites of startups from around the world. To accomplish this task, a web crawler was employed to visit each company's page and extract the keywords contained therein. Keywords were obtained from the HTML meta-keywords tag, as well as from the business names and other useful text data.

We opted to use meta-keywords in our study for several reasons. Despite their diminished importance in search engine rankings, many websites continue to employ meta-keywords for various purposes, such as internal search and content organization. These keywords can offer valuable insights into a website's content and focus. Meta-keywords present a concise and structured representation of the main topics covered by a website, making them a suitable information source for our word selection method. In instances where meta-keywords were unavailable or insufficient, our dataset was supplemented with alternative information sources, such as page titles, headings, and descriptions.

After collecting the raw dataset, a data cleansing process was conducted, which is discussed below. Non-English keywords were excluded during the data cleaning. To achieve this, natural language processing models were utilized to determine the language of the text segments. Additionally, parts of the keywords identified as irrelevant, such as "ltd", "org", "com", and others, were removed. Keywords and titles shorter than four and longer than 20 characters were also eliminated. These rules were established empirically by analyzing the raw data and training initial models, which allowed us to assess how the dataset should be constructed and identify any issues arising while generating different business names (texts). The following section provides examples of the data used in the study.

The following are examples of data that were used in the study.

Input	Output
food, juice bar, specialty food	For Goodness Shake
auto detailing, auto glass service, repair, auto	A Zone Auto Glass
resort, beauty, gift	The Phoenician Spa

Table 1. Example of input and output values for observations in a data set

3.2 Methods

In this section, we describe the models used in our study, along with the specific parameters and activation functions employed. The Generative Pre-trained Transformer (GPT) models utilize the Gaussian Error Linear Unit (GELU) activation function [22]. GELU is related to stochastic regularization techniques and is a modification of adaptive dropout [23]. In many cases, it is desirable for neural networks

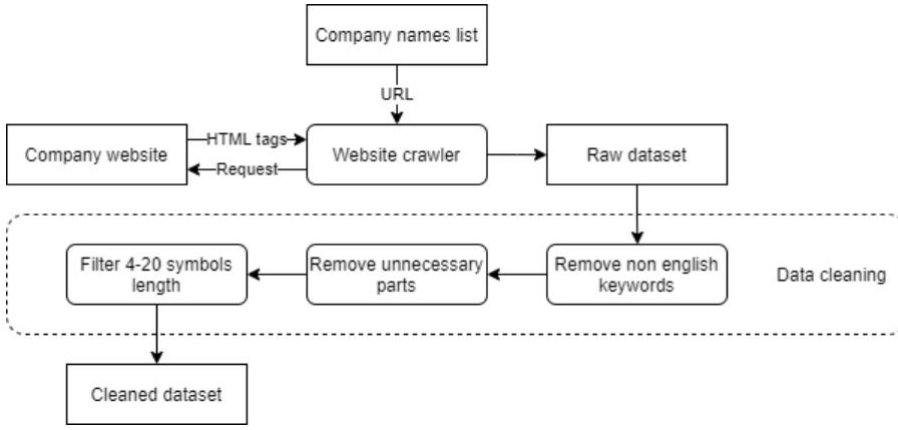


Figure 2. Schematic of data preparation process

to provide deterministic solutions, which necessitates the incorporation of nonlinearity in the network architecture. GPT models achieve this through the use of the GELU activation function, which adds the required nonlinearity while also preserving certain linear properties. The GELU activation function is defined as follows:

$$GELU(x) = xP(X \leq x) = x\phi(x) = x\frac{1}{2}\left[1 + erf\left(\frac{x}{\sqrt{2}}\right)\right], \tag{4}$$

which can be approximated by a simpler formula,

$$0.5x\left(1 + \tanh\left[\sqrt{\frac{2}{\pi}}\left(x + 0.044715x^3\right)\right]\right). \tag{5}$$

The choice of activation function plays a crucial role in the performance of neural networks, as it governs the flow of information through the network and influences the learning dynamics. By employing GELU, the GPT models can effectively learn complex patterns and representations in the input data, thereby enabling the generation of coherent and contextually relevant output. In the following subsections, we will detail the specific GPT models used in our study, outline their architectural differences, and discuss the parameter settings for each model during training and evaluation phases.

In this study, we utilize modifications of various GPT models. The key parameters characterizing GPT models include:

1. Maximum sequence length that can be processed by the models.
2. Encoder and Pooler layer dimensions (hidden size).
3. The number of attention heads in each transformer layer encoder.

4. The number of hidden layers in the transformer encoder.
5. Dropout probabilities (fully connected layer, embedding layer, attention layer).

Table 2 presents the parameters for the different GPT models used in our study. While this table does not provide information on dropout probabilities, initial range, and layer norm epsilon, these values are consistent across all GPT models: total dropout probability is 0.1; initial range is 0.02; and layer norm epsilon is 10^{-5}. Other parameters are detailed in the table.

It is worth noting that the original GPT model utilizes a vocabulary of 40 478 tokens, whereas the second version (GPT-2) expands the vocabulary to 50 257 tokens. GPT Neo models share the same initial range and vocabulary size as GPT-2. Interestingly, GPT Neo models do not employ dropout, and their global and local attention layers alternate in the following pattern: global, local, global, local, and so on. This unique configuration may contribute to the distinct performance characteristics observed in GPT Neo models compared to their GPT and GPT-2 counterparts.

Model	nhead	nlayer	nctx	nembd	npositions
GPT	12	12	512	768	512
DistilGPT2	12	6	1 024	768	1 024
GPT2	12	12	1 024	768	1 024
GPT2-Medium	16	24	1 024	1 024	1 024
GPT2-Large	20	36	1 024	1 280	1 024
GPT2-XL	25	48	1 024	1 600	1 024
GPT2-Medium	14.7/14.8	2:42	8.18	42.23	12.45
GPT2-Large	22.7/14.9	7:27	10.86	45.66	11.08
GPT2-XL	34.8/14.9	25:44	17.62	42.71	11.41

Table 2. Parameters for different GPT models

Model	Attention	Heads	Layers	Hidden Size
GPTNeo-125M	6	12	12	768
GPTNeo-1.3B	12	16	24	2 048
GPTNeo-2.7B	16	20	32	2 560

Table 3. Parameters for different GPT Neo models

It is important to note that some of the models in this study necessitate high-performance graphics card configurations. In our case, we utilized Nvidia T4 graphics cards. To further accelerate computations and efficiently manage distributed computing during model training, we employed the DeepSpeed library, developed by Facebook for Python.

DeepSpeed is a deep learning optimization library designed to streamline the use of distributed computing resources in model training. DeepSpeed leverages the

Zero Redundancy Optimizer (ZeRO) optimization strategies (as illustrated in Figure 3). These strategies eliminate redundant memory consumption across parallel data processes by partitioning the three model states (optimizer states, gradients, and parameters) among the processes instead of replicating them. This approach enhances memory efficiency compared to traditional data parallelism while preserving computational precision and communication efficiency.

In this study, we employed two DeepSpeed optimization strategies: ZeRO2 and ZeRO3. In the ZeRO2 stage, reduced 32-bit gradients for updating model weights are further divided, with each process maintaining only the gradients corresponding to a portion of its optimizer states. In the ZeRO3 stage, 16-bit model parameters are distributed across all processes. ZeRO-3 automatically assembles and disassembles these parameters as needed, further optimizing memory usage and training efficiency.

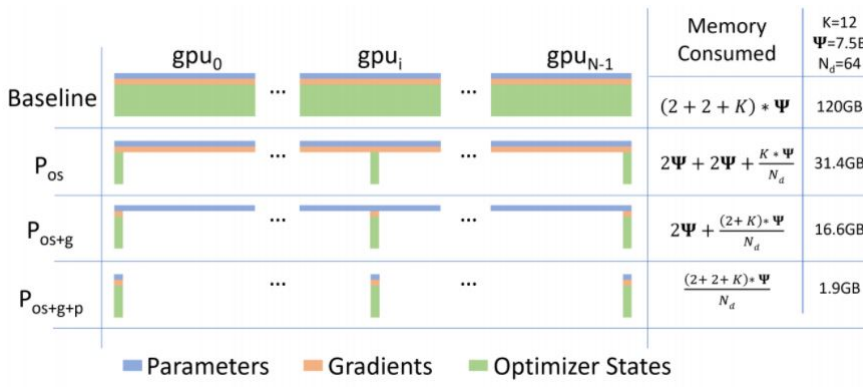


Figure 3. Comparing the per-device memory consumption of model states, with three stages of ZeRO-DP optimizations ([24])

Models that are not publicly available and were therefore not included in the study for this reason.

3.3 Experimental Setup

To compare different models, the data set was divided into two parts. The training data set represented 80 % and the test data set 20 %. The same data sets were used to train all individual models. The experiments in this study were performed using a Google Cloud Platform virtual machine with parameters of: 12 vCPUs, 78 GB memory, GPU: 1 × NVIDIA Tesla T4. For the largest models, the GPT-J-6B and GPT2-XL virtual machine parameters have been increased to 16 vCPUs, 150 GB of memory and 2 × NVIDIA Tesla T4.

3.4 Model Performance Evaluation

Models can be compared using perplexity metrics, which are calculated at the end of model development. Perplexity is a metric that measures the accuracy of probabilistic models, such as natural language generation models. It can be used to determine how well a generated sentence aligns with the sentences in the training sample [16, 25]. Perplexity is particularly popular for evaluating language model (LM)-based natural language generation models due to its ability to capture the diversity of generated text, which is crucial for tasks such as story generation [26], question generation [27], and question-answer generation [28]. However, it is important to note that perplexity is not a perfect metric for measuring text diversity, as a high perplexity does not necessarily imply low diversity. For instance, an LM with evenly distributed vocabulary for each generated token may exhibit high diversity but poor quality, resulting in a large perplexity value [29].

Assessing the performance of natural language generation systems solely based on perplexity may not always provide definitive conclusions. Traditionally, human evaluation has been used to assess the quality of generated text [30]. In such evaluations, subjects are shown generated text alongside human-written text and asked to compare them [31]. In some cases, subjects are presented with text generated by multiple systems for comparison. This methodology was first used in natural language generation (NLG) research in the mid-1990s by [32] and [33], and its popularity continues to this day. Human evaluation remains an essential criterion for assessing the accuracy of natural language generation models [34].

Research on language generation shows that different scales are used to assess language quality, such as Likert or continuous variable scales. Continuous scales provide evaluators the flexibility to assess at intermediate levels ([35, 36]). However, most evaluators prefer continuous scales over discrete ones. Despite the additional information provided by continuous variable scales, discrete Likert scales are more commonly used to evaluate generated language [37]. This preference is due to the difficulty respondents may face when evaluating using continuous scales. [38] show that up to 63 % of natural language generation articles use the Likert scale, as detailed by [39].

In our study, we used a discrete 7-point Likert scale, where a rating of 1 indicates that the generated name is inappropriate and the respondent would not use such a name, while a rating of 7 indicates that the generated name is highly appropriate and would be used by the respondent. We recruited 158 participants who were asked to evaluate business names generated by the models based on a given scenario: *"Imagine that your grandmother gave you her secret pancake recipe and you want to open your own business. But you have no idea how you should name your business. You found out that AI can offer you your business name with just a few words. Since you want to sell pancakes, you enter the words "Pancakes, Bakery, Shop" and AI gives you possible business names. And now it is your job to evaluate these names."*. The participants evaluated individual business names without knowing which model generated them. To avoid bias, we mixed the names generated by the various models

to ensure objective evaluation. Additionally, we included fine-tuned OpenAI models for comparison. To reduce the impact of extreme ratings, we excluded the best and worst evaluated models from our analysis.

4 RESULTS

This section presents the results obtained during our study. As the primary objective was to evaluate different text generation models, we assessed these models using both Perplexity metrics and human evaluation. According to the Perplexity metrics, the highest-rated model is the original GPT. However, when considering only the newer generation models, the best result is observed with the GPT-2 Medium model.

Interestingly, the study's results reveal a discrepancy between human evaluation and Perplexity assessment. Human evaluation shows that the best performance is achieved using the GPT-Neo-1.3B model, with its evaluation being statistically significantly higher compared to other models ($p < 0.05$). In contrast, the GPT-Neo-2.7B model exhibits inferior results, and its evaluation does not show a statistically significant difference from the GPT-Neo-125M model ($p > 0.05$), despite the latter being 20 times smaller.

These findings highlight the importance of considering multiple evaluation methods when assessing text generation models, as different metrics may provide distinct insights into a model's performance and capabilities.

Model	Peak RAM/VRAM Usage (GB)	Fine-tuning Time (hh:mm)	Perplexity	Avg. Score	Std. Deviation
Ada	–	–	–	41.81	10.05
Babbage	–	–	–	37.86	9.35
Curie	–	–	–	35.92	8.07
GPT	10.6/15	1:53	2.46	38.88	10.76
DistilGPT2	9.5/15	0:26	9.49	39.67	10.61
GPT2	10.5/14.5	0:46	10.26	43.25	11.42
GPT2-Medium	14.7/14.8	2:42	8.18	42.23	12.45
GPT2-Large	22.7/14.9	7:27	10.86	45.66	11.08
GPT2-XL	34.8/14.9	25:44	17.62	42.71	11.41
GPTNeo-125M	10.6/15	1:03	9.12	44.66	10.75
GPTNeo-1.3B	31.7/14.8	10:17	36.37	46.6	11.36
GPTNeo-2.7B	49/12.7	34:05	41.62	44.93	9.81
GPT-J-6B	101/15	72:25	37.08	–	–
XLNetBase	10.6/15	3:51	–	–	–
XLNetLarge	15.2/15	11:11	–	–	–

Table 4. Research models results: RAM/VRAM usage, fine-tuning, perplexity and average human evaluation scores

A critical aspect of using the ZeRO3 optimizer is its high RAM usage. Table 5 presents information on RAM consumption during the training of different models. The highest RAM usage is observed in the largest model, GPT-J-6B, reaching as much as 101 GB. Interestingly, GPT-2 XL and GPT-Neo-1.3B exhibit quite similar RAM usage patterns. Notably, the original GPT model consumes more RAM compared to GPT-2 and DistilGPT2.

Another objective of this study was to determine the maximum batch size for different text generation models using a single NVIDIA T4 graphics card. Our results showed that a batch size of up to 26 could be employed during DistilGPT2 training. In contrast, the largest models allowed for a maximum batch size of only 2, which complicates their utilization due to significantly longer training times.

To evaluate the text generation speed of various models, we performed 1 000 text generations, each generating five business names with a maximum length of 60 characters. We found that the fastest generation occurred using the DistilGPT2 model, with a generation time of only 0.49 seconds. Conversely, the slowest generation took place using the GPT model, which required 12.2 seconds. However, the GPT model exhibited the lowest coefficient of variation, indicating that it generates text consistently over a stable time period. These findings underscore the trade-offs between model size, training time, and generation speed when selecting an appropriate text generation model for a given application.

Model	Mean	Standard Deviation	Coefficient of Variation
GPTNeo-125M	1.046	0.385	0.368
GPTNeo-1.3B	5.684	1.424	0.251
GPTNeo-2.7B	9.335	1.521	0.163
DistilGPT2	0.490	0.190	0.388
GPT2-Medium	1.983	0.592	0.298
GPT2-Large	4.157	1.105	0.266
GPT2-XL	8.451	2.072	0.245
GPT2	0.825	0.303	0.368
GPT	12.211	0.673	0.055

Table 5. Research models inference speed (seconds) based on 5 sequences max length 60 generation for 1000 samples

The Friedman test is a non-parametric statistical test developed by Milton Friedman. Similar to the parametric repeated measures ANOVA, it is employed to detect differences in treatments across multiple test attempts. The procedure involves ranking each row (or block) together, then examining the values of ranks by columns. Applicable to complete block designs, the Friedman test is a special case of the Durbin test. Kendall's W Test refers to the normalization of the Friedman statistic and is used to assess the degree of agreement among respondents.

In our study, the Friedman ranks for the models were as follows: Ada (6.61), Babbage (4.08), Curie (3.63), DistilGPT2 (4.61), GPT (4.63), GPT-2 (7.02), GPT-2 Large (9.03), GPT-2 Medium (5.73), GPT-2 XL (6.70), GPT-Neo-1.3B (9.22),

	Ada	Babbage	Curie	DGPT2	GPT	GPT2	GPT2-L	GPT2-M	GPT2-XL	GPTNeo M	GPTNeo S
Babbage	−4.99*										
Curie	−4.99	−1.31									
Distil GPT2	−2.75**	−2.93**	−2.95**								
GPT	−3.36***	−1.369	−1.43	−1.02							
GPT2	−1.67	−5.31***	−5.23***	−4.31***	−3.57***						
GPT2-L	−4.82***	−6.96	−5.86***	−6.70***	−5.54***	−3.67***					
GPT2-M	−0.38	−4.45***	−3.66***	−3.67***	−3.33***	−1.79	−4.04***				
GPT2-XL	−1.07	−5.37***	−4.99***	−4.84***	−3.03***	−0.26	−4.66***	−1.51			
GPTNeo M	−5.32***	−6.93***	−5.60***	−6.04***	−6.43***	−4.80***	−2.13*	−4.33***	−4.42***		
GPTNeo S	−3.54***	−6.37***	−4.96***	−5.19***	−5.22***	−2.27*	−1.74	−3.78***	−3.38**	−3.69**	
GPTNeo L	−4.05***	−6.64***	−5.82***	−5.68***	−5.30***	−3.37**	−0.12	−2.85**	−1.33	−0.60	−2.03*

Notes: * $p < .05$, ** $p < .01$, *** $p < .001$

Table 6. Table of results of statistical differences between different models

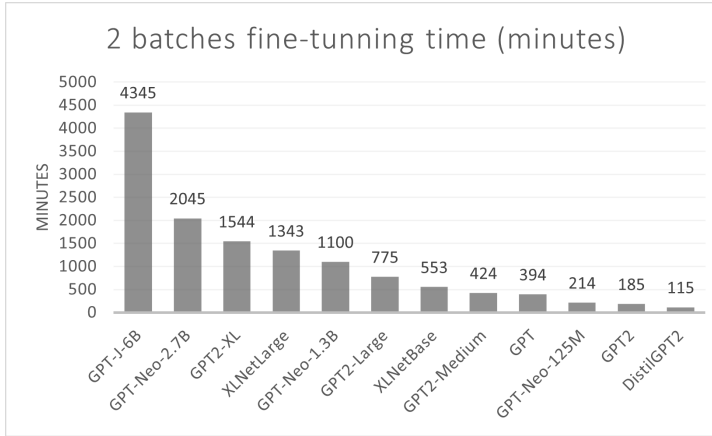


Figure 4. Comparison of fine-tuned models with 2 batches training time (minutes)

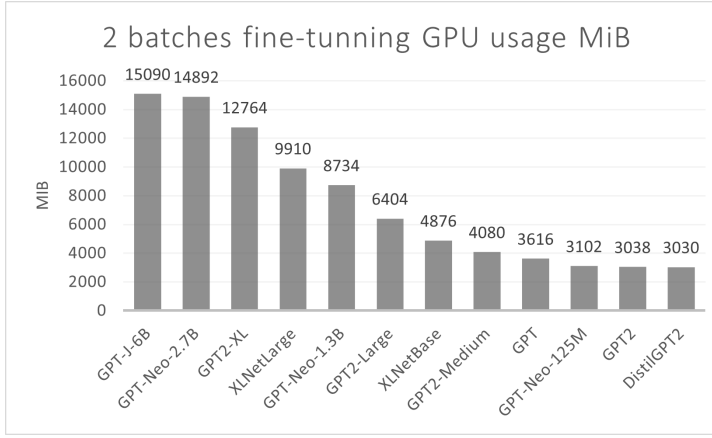


Figure 5. Comparison of fine-tuned models with 2 batches training VRAM usage (MiB)

GPT-Neo-125M (8.05), and GPT-Neo-2.7B (8.69). With a Chi-Square value of 181.335 and a p-value lower than 0.05, we can conclude that at least one of the models has a statistically significant difference from the others. Kendall's W value is 0.311.

Table 6 presents a pairwise comparison of the individual models. Interestingly, the GPT-Neo-1.3B and GPT-Neo-2.7B models do not exhibit a statistically significant difference in solving the business name generation problem ($p > 0.05$). Additionally, upon evaluating the OpenAI models, we observe that the Babbage and Curie models also do not differ statistically significantly ($p > 0.05$). These findings highlight the nuances in model performance and the importance of considering

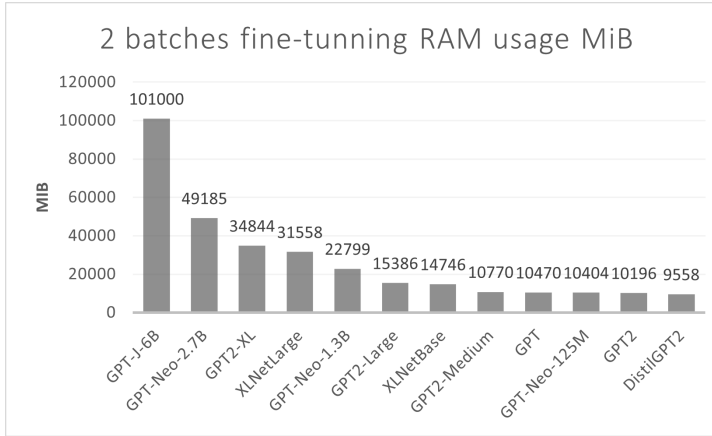


Figure 6. Comparison of fine-tuned models with 2 batches training RAM usage (MiB) with ZeRO3 optimization strategy

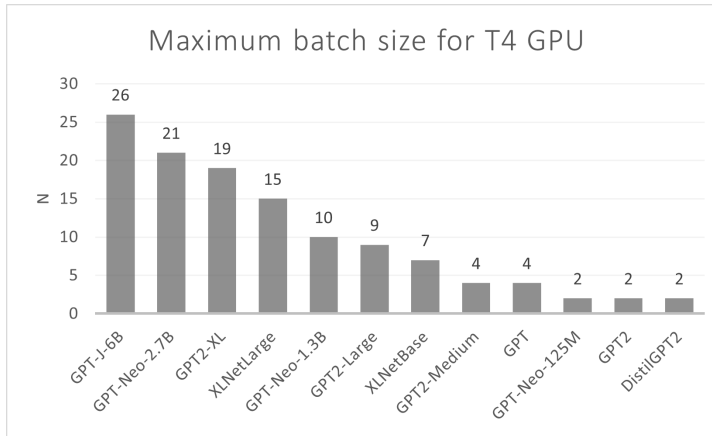


Figure 7. Comparison of fine-tuned models maximum batch size with one NVIDIA T4 GPU

multiple evaluation criteria when selecting an appropriate text generation model for a specific task.

5 CONCLUSIONS

This paper provides an overview of the transformer architecture and the primary transformer models currently available for use in natural language processing tasks. We also offer insights into the training process for particularly large models and present the idea of adapting natural language generation for business name genera-

tion, with further developments in this domain. The results of our study reveal that relying solely on the Perplexity metric does not always identify the best performing model. Human evaluation is a particularly important assessment method for natural language generation tasks, prompting us to conduct a consumer survey in our study. The findings demonstrate that, in the context of business name generation, larger models do not yield statistically significantly better results compared to their smaller counterparts. Consequently, employing larger models in practice may not be advantageous, as their name generation takes a statistically significant longer time than that of smaller models. Moreover, we observed that the newer generation of transformer models exhibits superior performance in generating business names, while XLNet models were not well-suited for this task. This highlights the importance of selecting appropriate models for specific applications and considering multiple evaluation criteria to ensure optimal performance and efficiency.

REFERENCES

[1] Merchant, K.—Pande, Y.: NLP Based Latent Semantic Analysis for Legal Text Summarization. 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2018, pp. 1803–1807, doi: 10.1109/ICACCI.2018.8554831.

[2] Yang, L.—Li, Y.—Wang, J.—Sherratt, R. S.: Sentiment Analysis for E-Commerce Product Reviews in Chinese Based on Sentiment Lexicon and Deep Learning. IEEE Access, Vol. 8, 2020, pp. 23522–23530, doi: 10.1109/ACCESS.2020.2969854.

[3] Dang, N. C.—Moreno-García, M. N.—De la Prieta, F.: Sentiment Analysis Based on Deep Learning: A Comparative Study. Electronics, Vol. 9, 2020, No. 3, Art. No. 483, doi: 10.3390/electronics9030483.

[4] Mishev, K.—Gjorgjevikj, A.—Vodenska, I.—Chitkushev, L. T.—Trajanov, D.: Evaluation of Sentiment Analysis in Finance: From Lexicons to Transformers. IEEE Access, Vol. 8, 2020, pp. 131662–131682, doi: 10.1109/ACCESS.2020.3009626.

[5] Xia, Y.—He, T.—Tan, X.—Tian, F.—He, D.—Qin, T.: Tied Transformers: Neural Machine Translation with Shared Encoder and Decoder. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 5466–5473, doi: 10.1609/aaai.v33i01.33015466.

[6] De Coster, M.—D'Oosterlinck, K.—Pizurica, M.—Rabaey, P.—Verlinden, S.—Van Herreweghe, M.—Dambre, J.: Frozen Pretrained Transformers for Neural Sign Language Translation. In: Shterionov, D. (Ed.): Proceedings of the 1st International Workshop on Automatic Translation for Signed and Spoken Languages (AT4SSL). Association for Machine Translation in the Americas, 2021, pp. 88–97, https://aclanthology.org/2021.mtsummit-at4ssl.10.

[7] Wolf, T.—Debut, L.—Sanh, V.—Chaumond, J.—Delangue, C.—Moi, A. et al.: Transformers: State-of-the-Art Natural Language Processing. Proceedings of

the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP), ACL, 2020, pp. 38–45, doi: 10.18653/v1/2020.emnlp-demos.6.

[8] Lu, Y.—Zhang, J.—Zeng, J.—Wu, S.—Zong, C.: Attention Analysis and Calibration for Transformer in Natural Language Generation. IEEE/ACM Transactions on Audio, Speech, and Language Processing, Vol. 30, 2022, pp. 1927–1938, doi: 10.1109/TASLP.2022.3180678.

[9] Vaswani, A.—Shazeer, N.—Parmar, N.—Uszkoreit, J.—Jones, L.—Gomez, A. N.—Kaiser, L.—Polosukhin, I.: Attention Is All You Need. In: Guyon, I., Von Luxburg, U., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.): Advances in Neural Information Processing Systems 30 (NIPS 2017). Curran Associates, Inc., 2017, pp. 5998–6008, doi: 10.48550/arXiv.1706.03762.

[10] Radford, A.—Wu, J.—Amodei, D.—Amodei, D.—Clark, J.—Brundage, M.—Sutskever, I.: Better Language Models and Their Implications. OpenAI Blog, 2019, https://openai.com/blog/better-language-models.

[11] Brown, T.—Mann, B.—Ryder, N.—Subbiah, M.—Kaplan, J. D.—Dhariwal, P. et al.: Language Models Are Few-Shot Learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., Lin, H. (Eds.): Advances in Neural Information Processing Systems 33 (NeurIPS 2020). Curran Associates, Inc., 2020, pp. 1877–1901, doi: 10.48550/arXiv.2005.14165.

[12] Radford, A.—Wu, J.—Child, R.—Luan, D.—Amodei, D.—Sutskever, I. et al.: Language Models Are Unsupervised Multitask Learners. OpenAI Blog, 2019.

[13] Devlin, J.—Chang, M. W.: Open Sourcing BERT: State-of-the-Art Pre-Training for Natural Language Processing. Google AI Blog, 2018.

[14] Sanh, V.—Debut, L.—Chaumond, J.—Wolf, T.: DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. CoRR, 2019, doi: 10.48550/arXiv.1910.01108.

[15] Lan, Z.—Chen, M.—Goodman, S.—Gimpel, K.—Sharma, P.—Soricut, R.: ALBERT: A Lite BERT for Self-Supervised Learning of Language Representations. CoRR, 2019, doi: 10.48550/arXiv.1909.11942.

[16] Lewis, M.—Liu, Y.—Goyal, N.—Ghazvininejad, M.—Mohamed, A.—Levy, O.—Stoyanov, V.—Zettlemoyer, L.: BART: Denoising Sequence-to-Sequence Pre-Training for Natural Language Generation, Translation, and Comprehension. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J. (Eds.): Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. ACL, 2020, pp. 7871–7880, doi: 10.18653/v1/2020.acl-main.703.

[17] Adhikari, A.—Ram, A.—Tang, R.—Lin, J.: DocBERT: BERT for Document Classification. CoRR, 2019, doi: 10.48550/arXiv.1904.08398.

[18] Liu, Y.—Ott, M.—Goyal, N.—Du, J.—Joshi, M.—Chen, D.—Levy, O.—Lewis, M.—Zettlemoyer, L.—Stoyanov, V.: RoBERTa: A Robustly Optimized BERT Pretraining Approach. CoRR, 2019, doi: 10.48550/arXiv.1907.11692.

[19] Dai, Z.—Yang, Z.—Yang, Y.—Carbonell, J.—Le, Q. V.—Salakhutdinov, R. R.: Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. CoRR, 2019, doi: 10.48550/arXiv.1901.02860.

[20] Yang, Z.—Dai, Z.—Yang, Y.—Carbonell, J.—Salakhutdinov, R. R.—Le, Q. V.: XLNet: Generalized Autoregressive Pretraining for Language Understanding. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (Eds.): Advances in Neural Information Processing Systems 32 (NeurIPS 2019). 2019, pp. 5753–5763, doi: 10.48550/arXiv.1906.08237.

[21] Gautam, A.—Venktesh, V.—Masud, S.: Fake News Detection System Using XLNet Model with Topic Distributions: CONSTRAINT@AAAI2021 Shared Task. In: Chakraborty, T., Shu, K., Bernard, H. R., Liu, H., Akhtar, M. S. (Eds.): Combating Online Hostile Posts in Regional Languages During Emergency Situation (CONSTRAINT 2021). Springer, Cham, Communications in Computer and Information Science, Vol. 1402, 2021, pp. 189–200, doi: 10.1007/978-3-030-73696-5_18.

[22] Hendrycks, D.—Gimpel, K.: Gaussian Error Linear Units (GELUs). CoRR, 2016, doi: 10.48550/arXiv.1606.08415.

[23] Ba, J.—Frey, B.: Adaptive Dropout for Training Deep Neural Networks. In: Burges, C. J., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K. Q. (Eds.): Advances in Neural Information Processing Systems 26 (NIPS 2013). Curran Associates, Inc., 2013, pp. 3084–3092, https://proceedings.neurips.cc/paper_files/paper/2013/file/7b5b23f4aadf9513306bcd59afb6e4c9-Paper.pdf.

[24] Rajbhandari, S.—Rasley, J.—Ruwase, O.—He, Y.: ZeRO: Memory Optimizations Toward Training Trillion Parameter Models. SC20: International Conference for High Performance Computing, Networking, Storage and Analysis, IEEE, 2020, pp. 1–16, doi: 10.1109/SC41405.2020.00024.

[25] Salvagno, M.—Taccone, F. S.—Gerli, A. G. et al.: Can Artificial Intelligence Help for Scientific Writing? Critical Care, Vol. 27, 2023, No. 1, Art. No. 75, doi: 10.1186/s13054-023-04380-2.

[26] Li, J.—Tang, T.—Zhao, W. X.—Wen, J. R.: Pretrained Language Models for Text Generation: A Survey. CoRR, 2021, doi: 10.48550/arXiv.2105.10311.

[27] Pan, L.—Lei, W.—Chua, T. S.—Kan, M. Y.: Recent Advances in Neural Question Generation. CoRR, 2019, doi: 10.48550/arXiv.1905.08949.

[28] Fan, A.—Jernite, Y.—Perez, E.—Grangier, D.—Weston, J.—Auli, M.: ELI5: Long Form Question Answering. In: Korhonen, A., Traum, D., Màrquez, L. (Eds.): Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. ACL, 2019, pp. 3558–3567, doi: 10.18653/v1/P19-1346.

[29] Tevet, G.—Berant, J.: Evaluating the Evaluation of Diversity in Natural Language Generation. Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, ACL, 2021, pp. 326–346, doi: 10.18653/v1/2021.eacl-main.25.

[30] Mellish, C.—Dale, R.: Evaluation in the Context of Natural Language Generation. Computer Speech and Language, Vol. 12, 1998, No. 4, pp. 349–373, doi: 10.1006/csla.1998.0106.

[31] Jones, K. S.—Galliers, J. R.: Evaluating Natural Language Processing Systems: An Analysis and Review. Springer, Berlin, Heidelberg, 1995, doi: 10.1007/BFb0027470.

[32] Coch, J.: Evaluating and Comparing Three Text-Production Techniques. Proceed-

ings of the 16th Conference on Computational Linguistics - Volume 1 (COLING 1996), ACL, 1996, pp. 249–254, doi: 10.3115/992628.992673.

[33] Lester, J. C.—Porter, B. W.: Developing and Empirically Evaluating Robust Explanation Generators: The KNIGHT Experiments. Computational Linguistics, Vol. 23, 1997, No. 1, pp. 65–101.

[34] Gkatzia, D.—Mahamood, S.: A Snapshot of NLG Evaluation Practices 2005 - 2014. Proceedings of the 15th European Workshop on Natural Language Generation (ENLG), ACL, 2015, pp. 57–60, doi: 10.18653/v1/w15-4708.

[35] Graham, Y.—Baldwin, T.—Moffat, A.—Zobel, J.: Continuous Measurement Scales in Human Evaluation of Machine Translation. Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse, ACL, 2013, pp. 33–41, https://aclanthology.org/W13-2305.pdf.

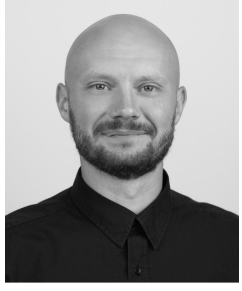
[36] Bojar, O.—Chatterjee, R.—Federmann, C.—Graham, Y.—Haddow, B. et al.: Findings of the 2017 Conference on Machine Translation (WMT17). Proceedings of the Second Conference on Machine Translation (WMT17), ACL, 2017, pp. 169–214, doi: 10.18653/v1/W17-4717.

[37] Gatt, A.—Krahmer, E.: Survey of the State of the Art in Natural Language Generation: Core Tasks, Applications and Evaluation. Journal of Artificial Intelligence Research, Vol. 61, 2018, pp. 65–170, doi: 10.1613/jair.5477.

[38] Amidei, J.—Piwek, P.—Willis, A.: The Use of Rating and Likert Scales in Natural Language Generation Human Evaluation Tasks: A Review and Some Recommendations. In: van Deemter, K., Lin, C., Takamura, H. (Eds.): Proceedings of the 12th International Conference on Natural Language Generation (INLG 2019). ACL, 2019, pp. 397–402, doi: 10.18653/v1/W19-8648.

[39] Joshi, A.—Kale, S.—Chandel, S.—Pal, D. K.: Likert Scale: Explored and Explained. British Journal of Applied Science and Technology, Vol. 7, 2015, No. 4, pp. 396–403.

Mantas Lukauskas is a doctoral student in computer science at the Department of Applied Mathematics at the Kaunas University of Technology (KTU). Currently working as a data scientist at the Lithuanian company Hostinger/Zyro, and a researcher in the projects funded by the Kaunas University of Technology and the Research Council of Lithuania. The main area of interest is artificial intelligence/machine learning (natural language processing, clustering, classification, and others) and its application in different practice areas like economics, trade, logistics, e-business, and healthcare.



Tomas Rasymas received his B.Sc. and M.Sc. degrees from the Vilnius University, Lithuania, in 2005 and 2007, respectively. His research interests include NLP, speech recognition, and generative AI.

Matas Minelga received his B.Sc. degree in computer science from the Kaunas University of Technology (KTU), Kaunas, Lithuania, in 2019. He applies his knowledge in machine learning to practical real-world use cases. His work is characterized by an innovative approach to leveraging machine learning algorithms when addressing critical challenges in technology-driven sectors. With a robust academic foundation and a dedicated focus on computational solutions, Matas continues to explore advanced techniques and methodologies in his area of work, consistently seeking opportunities to enhance efficiency and precision in machine learning systems.

Domas Vaitmonas received his B.Sc. from the Vilnius University in computational physics in 2016. He is currently a machine learning engineer at 10 speed.

Computing and Informatics, Vol. 42, 2023, 546–567, doi: 10.31577/cai_2023_3_546

NEW GAME-THEORETIC CONVOLUTIONAL NEURAL NETWORK APPLIED FOR THE MULTI-PURSUER MULTI-EVADER GAME

Nabila Sid, Meriem Djezzar, Mohammed El Habib Souidi, Mounir Hemam

ICOSI Laboratory
University of Khenchela, Algeria
e-mail: {sid.nabila, meriem.djezzar, souidi.mohammed, hemam.mounir}@univ-khenchela.dz

Abstract. Pursuit-Evasion Game (PEG) can be defined as a set of agents known as pursuers, which cooperate with the aim forming dynamic coalitions to capture dynamic evader agents, while the evaders try to avoid this capture by moving in the environment according to specific velocities. The factor of capturing time was treated by various studies before, but remain the powerful tools used to satisfy this factor object of research. To improve the capturing time factor we proposed in this work a novel online decentralized coalition formation algorithm equipped with Convolutional Neural Network (CNN) and based on the Iterated Elimination of Dominated Strategies (IEDS). The coalition is formed such that the pursuer should learn at each iteration the approximator formation achieving the capture in the shortest time. The pursuer's learning process depends on the features extracted by CNN at each iteration. The proposed supervised technique is compared through simulation, with the IEDS algorithm, AGR algorithm. Simulation results show that the proposed learning technique outperform the IEDS algorithm and the AGR algorithm with respect to the learning time which represents an important factor in a chasing game.

Keywords: Multi-agent system, Pursuit-Evasion Game (PEG), Convolutional Neural network (CNN), coalition formation

1 INTRODUCTION

In multi-agent systems (MAS), connected autonomous agents act in a limited environment to achieve objectives or to maximize rewards. MAS is a distributed system based on a set of agents that interact in most cases. The interaction is effectuated according to the mode of coordination used. This coordination can be represented through competition, cooperation, or either a negotiation between the agents. MAS have been mostly processed through the use of machine learning principles. In this paper, we propose a Convolutional Neural Network (CNN) to predict the approximate coalition according to the changes of the elimination priority. The CNN is effective in our case. CNN used to minimize error and maximize the probabilities to extract the indexes corresponding to the optimal coalition which make a capture in considerable time. The CNN reflects its simplicity and usefulness to model such problems. CNN consists of a multilayer stack of neurons, mathematical functions with several adjustable parameters, which preprocess the coalition's features. CNN categorized this features obtained from IEDS technique to generate the data set of input layer. A selection between several activation functions in our case was the Maxout function activation. This function activation is based on the fact that if k scalar products are provided for a node, effectively this node can learn a local nonlinear activation function by approximating it with a piecewise linear function consisting of k intervals. several researchs treated this field. In [1], the authors proposed to calculate the decentralized optimal strategy under uncertain environment in MAS. The learning in this case maintains five neural networks for each agent. Other authors [2], depend on principle of complete perception of environement for each agent, when he choose to use a recurent network to extract the feautures needed to make a partial perception of environment.

MAS has multiple uses in artificial inteligence, pursuit evasion and game theory in particular [3, 4, 5]. In the pursuit-evasion problem, there are many pursuit groups, each one contains a certain number of agents known as pursuers. Each pursuit group attempts to capture a specific evader in the shortest possible time. The multi-agent pursuit problem was processed by using several methods of coordination such as cooperation [6], MAS organizational models [7]. PEG was developed in the extensive research [8, 9, 10, 11, 12]. During the pursuit, the pursuers have to cooperate to form coalitions or pursuit groups. Coalitions are formed to capture evaders and dissolve after the task processing.

Game theory studies the options of autonomous agents as well as their consequences during the interactions. The most recent game-theoretic principle used to form pursuit coalition formation was the Iterated Elimination of Dominated Strategies model (IEDS). During each pursuit iteration in the IEDS algorithm [13], the pursuers follow a specific grouping strategy known as a coalition. Each pursuer aims to be a part of the coalition returning the maximum value extracted by the IEDS algorithm knowing that the coalitions of the pursuers are generated using static elimination priority.

IEDS was used in combination with Markov decision process (MDP) that allows the displacement of the agents. MDP is a mathematical framework used for modeling decision-making problems, where the outcomes are partially random and partially controllable [14].

In this paper, we propose a machine learning method, which used to predict the best coalition according to the changes of the elimination priority. In this case, the most appropriate machine learning method to utilize is Convolutional Neural Network (CNN). This latter, is a universal approximator for continuous functions within a bounded domain which is considered as a sub-class of neural networks, used to minimize error and maximize the probabilities in chasing game. The CNN reflects its simplicity and usefulness to model such problems. CNN consists of a multilayer stack of neurons, mathematical functions with several adjustable parameters, which preprocess small amounts of information. These CNN can categorize information extracted from IEDS algorithm to generate the data set of input layer. A selection between several activation functions in this kind of neural network is the Maxout function activation. This function activation is based on the fact that if k scalar products are provided for a node, effectively this node can learn a local non linear activation function by approximating it with a piecewise linear function consisting of k intervals.

A nother work proves the effectiveness of multiple maxout activation function variants on 18 datasets using Convolutional Neural Networks [15]. In [16], Cai and Liu combined maxout neurons with convolutional neural network (CNN) and the long short-term memory (LSTM) recurrent neural network (RNN).

In this work, we used the maxout to predict the maximum value used to select the optimal coalition formation in the shortest time where the selection of agents is effectue randomly, unlike IEDS algorithm wich given a previously priorities to agents to make coalitions, each coalition corresponds to a particular strategy. An elimination of dominated strategies is applied to extract the optimal coalition corresponding to the maximum value.

The structure of this paper is as follows: We discuss related research in Section 2. Section 3 describes the multi-agent pursuit problem and clarifies the principal characteristics of pursuers and evaders. In Section 4, we describe our proposed framework for pursuit MAS game equiped with CNN. The different steps of the proposed coalition formation algorithm are detailled in Section 5. Section 6 is dedicated to the presentation of experimental results. Finally, Section 7 concludes and provides directions for future work.

2 RELATED WORK

Lately, the pursuit-evasion problem has attracted the attention of many research activities in the area of multi-agent systems. Someof them depends on a new form of probability density function (PDF) to solve the optimal pursuit-evasion strategies and a neural network to estimates an optimal control, and approximates the optimal

cost function [17]. In another work, they designed an algorithm based on relaxation problem to estimate future states of the game by introducing a polynomial-time algorithm to control action selection in visibility [18].

Tian et al. [19], established a hierarchical evasion strategy for the Air-breathing Hypersonic Vehicles (AHVs). Specifically, the authors presented multiple cooperative pursuers in the case where they come from different directions concurrently. To achieve successful evasion, successive pursuers come from the same direction and flight with proper spacing. Others preferred to investigate the interaction between two antagonistic agents in an environment without obstacles [20]. Another interesting work [21] is based on the selection of an intelligent evader to be hunted by a group of pursuers by retreating horizon control policies.

Among the most processed problems in a multi-agent system, we can cite the Coalition Formation. Recently, in [22] Guo et al. have proposed a genetic algorithm with heuristic initialization and repair strategy (GAHIR) to solve coalition formation problem, they treated the problem as a single-task single-coalition formation, a multi-task single-coalition formation, as well as a multi-task multi-coalition formation. In the same issue, we find another research activity [23] that focuses on organizational hierarchies of coalition formation structures.

Furthermore, in [24] Estrada et al. addressed the problem of task allocation in Mobile Crowd Sensing (MCS) by forming tasks publisher coalition taking into consideration workers' route preferences. On the other hand, Babu and Chitnis [25] introduced the utility function to achieve the optimal coalition among nodes.

In comparison with other research activities, processing the same problem, Souidi et al. [26] introduced a coalition formation algorithm based on Agent-Group-Role organizational membership function model (AGRMF), which is an extension of agent-group-role (AGR) model. In this model, a group is considered as a fuzzy set where the goal always remains to optimize a coalition of agents in order to capture an evader in the shortest time. In another work [27], Cruz et al. have used reinforcement learning principles and smoothing techniques to modify the path planning of the agents in an unknown environment. These modifications have for objective to predict the greedy actions of pursuers.

In AGR organizational model [28], each agent can play one or more roles simultaneously. Each agent can be a member of one or more groups in the same time. A class of agents is determined by its task that is contributed to each entity of agents. The agents can effectuate a set of operations in the same group such as communication, cooperation, and negotiation. Benoudina et al. [29] used a multi-agent platform to build a simulator based on reactive agents capable to transform this complex system into a data processing program that can represent its structure, its communication, its behavior, its control loops and verify the integrity and its proper functioning.

Boudjidj et al. [30] have introduced a new proposal that consists on a transformation of Agent-Group-Role (AGR) organizational model in a categorical way, which permits the analysis, the verification, and the validation of the organization at a high level of abstraction.

Qadir et al. [31] affirmed, that in AGR model, there is no mechanism defining the access conditions regarding the groups. Consequently, in AGRMF model, they used a binary variable instead of logic fuzzy set called degree of membership. This degree of membership function controls whether the agents will take the role. A membership function interpreted within some parameters the options of an agent to undertake a role.

In [32], the authors used concept of AGR model to represent a liquefied natural gas treatment process (decarbonation) at a gas plant in Algeria. Moreover in [30], the authors prposed a transformation of Agent-Group-Role (AGR) organizational model in a categorical way in order to obtain a formal semantics model describing the MAS organization, which allows a high level of abstraction.

MDP is used in several domains, such in [33], the author devises vehicle trajectories by coupling a locally-optimal motion planner with a Markov decision process (MDP) model that can capture network-level information.

Recently, many research activities have taken intense interests of the features in neural networks.

Other researches, such as [34], have taken into consideration the performance of hyperparameter optimization of Deep CNNs by adapting Q-learning and defining learning agents per layer to split the design space into independent smaller design sub-spaces. Consequently, each agent can fine-tune the hyperparameters of the assigned layer concerning a global reward. A combination of graphic convolutional neural network with deep Q-network has been used in [35] to form an innovative graphic convolution Q network, that serves as the information fusion module and decision processor in multi-agent cooperative control of connected autonomous vehicles.

3 PROBLEM STATEMENT

According to Schenato et al. [35], the pursuit-evasion game is defined as *"a mathematical abstraction arising from numerous situations, which address the problem of controlling a swarm of autonomous agents in the pursuit of one or more evaders"*.

Much research in the pursuit evasion field focuses on the capture of evaders neglecting the factor of time. In our work, achieving the minimum possible capture time was our interest.

For this purpose, we depended on a supervised learning which benefits from exploiting the coalitions categorized by the IEDS technique with a convolutional neural network so that the MAS be able to self learn its forming coalition by adjusting its parameters (i.e. the weights of the neurons), so as to reduce the difference between the capture time obtained and the expected capture time.

The margin of error is thus reduced over the training processs, with the aim of being able to generalize one's learning to new cases. The weak point of this method is that it does not give all its predictive capacity when the input data are small. In other words, this technique gives good results as long as we have hyper data sets.

The point distinguishing this approach from the IEDS, is the implementation of a online prediction mechanism based on CNN.

Our work will depend on the performance provided by CNN using supervised learning. We depend on supervised learning or associative learning in which the network is trained by providing it with input and matching output patterns. Unlike the IEDS technique wich selected the agents with priority, in our case the agents are selected randomly without priority to forme the groups. We proposed two different types of evader, dynamic and static evader.

4 THE PROPOSED APPROACH

Forming efficient coalitions is one of the major research challenges in the area of multi-agent systems. In coalition formation, coherent sets of distinct, autonomous agents, interact to accomplish their individual or collective goals. A pursuit coalition begins with a task and dismisses when it is accomplished. A set of coalition extracted by IEDS technique is denoted by $S = \{c_1, c_2, \ldots, c_m\}$.

In order to decrease the communication cost and avoid repeated information in interactions, we based ourselves on the performance of CNN. The latter, uses the process of backpropagation to adjust the weights of neuron. A CNN has to be configured such that it can approximate at each iteation the optimal coalition which verify the condition: $(c_i, index_i) > (c_j, index_j)$. In other words, only the coalition having the maximum index, extracted by CNN, can be combined to the lowest number of iterations, wich means the shortest capturing time.

The agents playing the role *Pursuers* are denoted by $P = \{p_1, p_2, \ldots, p_n\}$. In addition, the *Evaders* are denoted by $E = \{e_1, e_2, \ldots, e_m\}$. The main goal of the pursuit-evasion game is to perform the capture of each evader by a group of pursuers in a finite time. Each evader is characterized by a degree of difficulty denoted D, $D = \{d_1, d_2, \ldots, d_n\}$, which represents the number of pursuers needed in a given pursuit. An evader e is defined by a type Re, with $Re \in \{I, II, III, IV\}$. This latter allows specifying how many pursuers are required to achieve the capture of the evader e.

The Pursuers achieving the capture of an evader e will get a reward equal to Re. The reward is provided to pursuers achieving the performance of the capture. The stability degree represents the roles' reattribution in a given coalition. It is assumed that all players have the same motion velocity (i.e. one cell per iteration), a stable communication system, and a partial vision of the pursuit region. In a bounded environment containing obstacles that pursuers must avoid, we consider the existence of eight (08) pursuers and two (02) evaders.

We perform our pursuit-evasion game in a rectangular two-dimensional grid with 100×100 cells. The agents and obstacles are located randomly in the grid of cells. Each cell corresponds to a specific state.

MDP application aims to compute the possibilities of the next state according to actual ones. This computation is effectuated in order to allow the pursuers' movements according to the detected reward.

Each neuron has its weight. To speed up the generation weights at the nodes, we used a random number generator. These numbers form a sequence of independent and uniformly distributed random variables on $[0, 1]$. This sequence is interpreted as the realization of a random variable which follows the law of uniform density on $[0, 1]$.

$$\mu_{i+1} = \pounds(\mu_1,\ \mu_{i-1}, \ldots, \mu_{i-k}), \quad k < i,\ i = 1, 2, \ldots, n, \tag{1}$$

where $\mu_{i+1} = \pounds(\mu_i)$, $\pounds$ has value $\in [0, 1]$.

The multi-agent system is used to learn new behaviors such that the natural systems exploit its performance in approximation method [34]. We intend to train our system to learn how to predict the optimal coalition making the best shortest time for a multi-agent pursuit game. This inspired us to believe that extracting for each coalition, the maximum average of reward and the stability degree as a input data can optimize the training process.

We assemble our training Dataset Dt through IEDS technique which resulting Reward (x_i) and stability degree (y_i) in each given coalition. Due to of large coalitions generated at each iteration, we are going to exploit the convolution layer in CNN. This later will contain samples of these coalitions in a vector to extract the maximum from its maxout units.

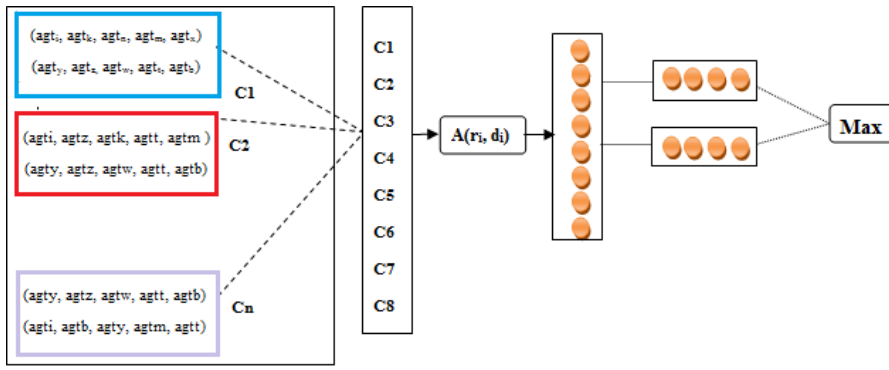


Figure 1. The CNN with tow unit maxout. A part of coalitions formation generated by IEDS technique are ranged in a vector. An average is extracted for each coalition. The tow Maxout unit calculated the maximum for each given coalition.

We use $(x_i, y_i) \in D$, $\forall i \in [1, n]$.

$$Dt = \begin{pmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \ldots & \ldots \\ x_n & y_n \end{pmatrix}, \tag{2}$$

where $x_i \in R^d$ is a d-dimensional sample, with each dimension corresponding to a particular value of the reward, and $y_i \in R^1$ is the stability degree.

4.1 Training of CNN

The challenge is to train the CNN by feeding it teaching patterns and letting it change its weights according to output to achieve a desired capture time. First, the calculation of the weighted product of the inputs (h_i) is effectuated according to the following expression:

$$\forall i \in [1, n] : h_i = (w_i \times z_i). \tag{3}$$

To normalize this product and avoid a drastically different range of values, we use what is called an activation function. An activation function transforms these values into values between $[0, 1]$ or $[-1, 1]$ to make the whole process statistically balanced. From this value, a transfer function calculates the value of the state of the neuron. This value will be transmitted to downstream neurons. There are many possible forms concerning the transfer function. Most of transfer functions are continuous, offering an infinity of possible values included in the interval $[0, +1]$ (or $[-1, +1]$).

4.2 Supervised Convolutional Neural Network Model (IEDSNN)

A novel model of supervised learning based Convolutional Neural Network is established. A supervised training is used to update weights of the network until the obtainment of minimum error. The error function will serve in constructing efficient supervised training algorithms to accelerate the learning process.

4.3 Supervised Learning

In supervised learning, the data entering the process are already categorized according to IEDS algorithm, which the proposed algorithm must use to predict an approximated outcome. Our algorithm will learn the input to output mapping function:

$$H = f(Z). \tag{4}$$

The goal is to understand the mapping function. In other words, when new input data (z_i) are introduced, we can predict the output variables (h) for that data. In supervised learning, the robustness of the algorithm will depend on the precision of its training. A supervised content learning algorithm produces an internal map that allows its reuse to classify new amounts of data.

Data preprocessing. This step allows calculating the input values, which are ranged in a matrix of two dimensions. Each dimension reflects a result extracted by IEDS algorithm. The average is calculated for all lines in the matrix. Each average will form a new value as input values of our input layer.

An average Z_i of two dimensions has been placed to represent the Input of our CNN:

$$\forall i \in [1, n] : Z_i = \frac{(x_i + y_i)}{2}. \tag{5}$$

Pre-training. Training the CNN requires specifying an initial value for the weights. A well-chosen initialization method will make learning easier. A distribution of random values is used to potential weights, we assign a constant number to all the weights. The constants numbers are in the range of $[-1, 1]$. The purpose of the random weight initialization is to break the symmetry. However, since the weights are no longer symmetrical, we can safely initialize all bias values with the same value. A well-chosen initialization can accelerate the convergence of the gradient descent, increasethe chance of gradient descent converging to lower training (and generalization) error.

Consequently, the parameters to initialize in our convolutional neural network are:

- Weight matrices

$$(W[1], W[2], W[3], \ldots, W[L-1], W[L])(Z[1], Z[2], Z[3], \ldots, Z[L-1], Z[L]).$$

- The bias vectors

$$(b[1], b[2], b[3]).$$

4.4 Function Activation Maxout

An activation function is a mathematical function used on a signal. It will replicate the activation potential found in the field of human brain biology. Moreover, it will allow the passage of information or not if the stimulation threshold is reached. Concretely, its role will be to predict whether or not to activate a neuron response. This prediction function that the CNN must learn is highly nonlinear. A neuron will only perform the following function:

$$z_i = \sum_{i=1}^{n} (input_i \times weight_i) + bias_i. \tag{6}$$

The Maxout activation function is chosen between severals activation function to capture the underlying nonlinearity. The Maxout activation function is a generalization of the Relu function [16].

The Maxout activation function is the most appropriate in our case. It is a piecewise linear function that returns the maximum of the inputs. Maxout activation function provides better optimization performance despite Castaneda et al. [21] are seen that in theory, a large number of extra parameters introduced by the k linear functions of each hidden Maxout unit result in large RAM storage memory cost and

considerable increase in training time, which affect the training efficient of very deep CNNs.

In general, a Maxout activation function is defined as follows:

$$h_i(x) = \max_{j\in[1,k]}(x_{ij}), \tag{7}$$

where

$$x_{ij} = x^T \times w_{ij} + b_{ij}. \tag{8}$$

$w \in R^{d\times m\times k}$ and $b_{ij} \in R^{m\times k}$ are the learned parameters, where m is the number of hidden units, d is the size of input vector and k represents the number of linear models. This nonlinearity can also be viewed as a feature selection process [24].

In our works, at each iteration by IEDS technique, a set of coalitions was generated resulting at the end of iteration a reward and degree of the maxout unit implements the following function:

$$H(x) = \max(unit_1, unit_2), \tag{9}$$

$$H(x) = \max(\max(W_1 \times z_1 + b_1, \ldots, W_n \times z_n + b_1), \tag{10}$$

$$\max(Z_1 \times z_1 + b_2, \ldots, Z_n \times z_n + b_2)). \tag{11}$$

The maxout-node applies n different scalar products to k offsets $(b1, b2)$ and finally takes the maximum of these n values. Such model will estimate the optimum coalition formation as follows:

$$h_i = CNN\left((W_i, w_i), (Z_i, z_i)\right). \tag{12}$$

4.5 Error Function

Error function isused to determine the performance of a neural network during learning. The derivative of the error function is used by iterative learning algorithms. We have the squared error such that:

$$(I_{p,k})^2 = |d_{p,k} - o_{p,k}|^2, \tag{13}$$

where p is the p^{th} form, d is the desired value, o is the obtained value.

We then seek W such that W has to be minimized. Descending gradient method (generalized delta rule) has been used. So, the weight W must change in the same direction as $(-\partial E/\partial W)$.

4.6 Training Process

Learning consists of training the convolutional neural network (CNN) to predict the approximate the shortest time needed to make a shortest time in capture. Tow groups of pursuers chasing tow evaders. During this chase, a set of parameters was

extracted by IEDDS technique. This prametters calculated for each given coalition are the number of changing role, wich represent the degree of stability of agents, number of iteration, reward.

For each coalition an average Zi was calculated for all iterations IEDS technique and ranged in a list. This list represent the input data for our CNN. The convolutionel layer will take at each time t a sample of this list and divided it into two lists: list1 [$Z1$ $Z2$ $Z3$ $Z4$] and list2 [$Z5$ $Z6$ $Z7$ $Z8$].

The Maxout unit will contain each given list. At this stage, an initialization step means to intialize the weights randomly, bias, threshold, learning rate. For each unit Maxout we extract the maximum value wich present the output: $h = \text{Max}(\text{Maxout unit1, Maxout unit2})$.

Unlike IEDS technique which is based on static percentages in the calculation of maximum value. The output value will serve to compare the coalition having the maximum couple of reward and degree of stability with the actual maximum wich used in capturing process.

We attempt to modify those weights according to the desired value with back-propagation algorithm. This algorithm is of the *online* type, when the weights are updated for each learning sample introduced to the neural network. Initially, the training process will propagate forward the inputs until obtaining an input calculated with the CNN. The second step consists of similitude between desired and calculated outcome. We adjust the weights such that in the next iteration the error must be minimized.

We propagate the signal forward in the layers of the CNN: $x_k^{(n-1)} \rightarrow x_j^{(n)}$

$$x_j^{(n)} = g^{(n)} \times v_j^{(n)} = g^{(n)} \times \sum_k w_{jk}^{(n)} x_k^{(n)}, \tag{14}$$

where g is the activation function Maxout and v_j is the agregation function.

Once the propagation is done we result our output value y. We can calculate the error between the y given by our CNN and the desired value t_i.

$$e_i^{output} = g(v_i^{output}[t_i - y_i]. \tag{15}$$

The weights updated as follow:

$$\Delta W_{ij}^{(n)} = e_j^{(n)} z_j^{(n-1)} \alpha, \tag{16}$$

where α is the learning rate, $(0 < \alpha \leq 1)$.

$$W_{ij} = W_{ij} + \Delta W_{ij}. \tag{17}$$

5 IEDSNN ALGORITHM

The decentralized coalition formation algorithm, that we proposed, is an extension of the Iterated Elimination of Dominated Strategies (IEDS) and equipped with

Convolutional Neural Network (CNN) to form dynamic pursuit groups. A potential coalitions are formed depending on performances of CNN. The process begins with set of agents making the role of pursuers selected randomly. For each coalition we extract the stability degree and reward. A prediction of maximum indicators of optimal coalition formation have been developed by CNN. In order to generate the coalition making the capture of evaders in the shortest time (lowest number of iteration), we implement the pseudo-code Algorithm 1.

Algorithm 1 IEDSNN

Input:
– t: the vector of features extracted from coalitions generated by IEDS technique
Output:
– Max ppredicted value leading to detect optimal coalition making a shortest chase
Begin
Lanch_Chase();
Calculate_Desired_Max();
while (($C_{life} > 0$) and not obstacle) **do**
Pursuit_Iteration ();
extraction_Feautures ();
Initializing indicators of CNN();
for i ← 1 to m **do**
$Inputnode_i \leftarrow T[i]$;
end for
Calculate_Error();
while (Max calculated by CNN $\leqslant$ Max desired) **do**
Calculate_Max();
Update_Wheights();
Propagate_Signal_Forward();
Propagate_Error_Forward();
end while;
index ← Max;
Extraction(coalition, index);
end while

A description for IEDSNN algorithm is summarized, in Figure 2, by a flowchart, when it started by localization of agents in a pursuit closed environment with 100×100 cells, 08 pursuers, and 2 evaders needing each one 4 pursuers to be blocked. By the fact that we extended our proposal of IEDS approach, we have depended on stability degree and reward value extracted from each coalition as a training data for CNN. After the training step using IEDS algorithm which is considered as a preliminary outcome, the IEDSNN starts training the pursuers to make a fast capture of evaders in the shortest time.

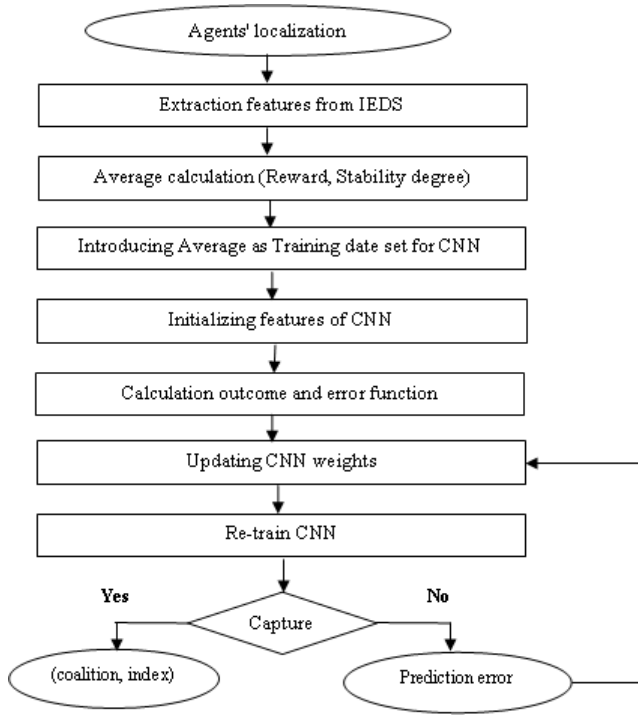


Figure 2. Flow chart of IEDSNN Algorithm

The process begins with the manual identification of the max value of the input vector; the vector containing the computation of the averages. This training iteratively decrease the error function which forces the algorithm to re-train the IEDSNN to finally extract the coalitions that have the highest contribution values to MAS efficiency.

First, the calculated averages are entered as input data. The phase of initialization of the CNN indicators took place. The training of the agents to form an appropriate coalition will be repeated as long as the least error function value is not yet achieved, implying that the expected coalition has not formed yet. Among all the coalitions carried out, the CNN extract the maximum value from each single unit which corresponds to any coalition. The training leads to perform an update of the weights always using the value of the error function obtained in each iteration.

Finaly, the trainig will stop when the predicted value is achieved, extracting in return the optimal coalition making the best chase in the shortest time. A several episode of chasing have been developed by this process, at each episode the number of coalition making the quick capture decrease as mentioned in Table 1. This decrease of number of coalition is definitive proof of the success of our proposal in anticipation of optimal coalition formation of agents in a given chase.

6 EXPERIMENTAL RESULTS

The operating flow of our CNN algorithm is summarized from the moment of recovering the computations values of the algorithm IEDS until the capture in a considerable time. Here we investigate the performance of the proposed algorithm for eight (08) pursuers and two (02) evaders of type $R_e = IV$.

IEDSNN algorithm exploits the performance of CNN specifically the activation function Maxout which reduces extracting the minimum time in a pursuit game.

Table 1 indicates the average capturing time as well as the average obtained payoff per pursuit iteration regarding the three compared approaches, AGR [26], IEDS [13], and the proposed IEDSNN.

	AGR	**IEDS**	**IEDSNN**
Average capturing time (iteration)	100.33	78.5	64.16
Average pursuers' rewards obtained by iteration	0.34	0.36	0.45

Table 1. Pursuit result

The new coordination mechanism applied imposes an equitable sharing of the tasks between the different pursuers. The showcased results reveal the crucial difference between them through the distinctive decrease in average capturing time of the IEDSNN approach which only makes 64.16 iterations in comparison with AGR (100.33 iterations) and IEDS (78.5 iterations). Our proposal, according to Table 1, shows the fast prediction of optimal coalition to realize a speed pursuit capture in comparison with the other approaches.

Figure 3 represents the pursuit capturing times obtained in 30 pursuit episodes regarding the three compared cases. A learning of IEDSNN approach during the pursuit of agents is shown by the average capturing time achieved.

Figure 3 reflects the ability of pursuers to get learned how to form optimal coalition formation that leads to capturing the evaders in considered time. The number of coalitions established by the pursuers at the first time is increased knowing that the system at the beginning is not learned. The first phase consists on depending on IEDS algorithm computations. Extracting the first chase with the values needed in IEDSNN algorithm as a pre-training phase in the process of training the whole IEDSNN. Then the pursuers started learning the flow of IEDSNN. The pursuers learn quickly in forming optimal coalitions which, in turn, leads to effectuate a quick pursuit capture. This is why the curve decreases from 99 iterations to 52 iterations and remained on average in the range of 60 to 52 iterations. This fact represents the main contribution of the proposed IEDSNN.

The main results, indicated in Figure 4, reflect the efficiency of IEDSNN algorithm regarding the average reward development and obtained per iteration as plotted in cube Figure 4 and Figure 5 in the three compared cases. These results prove the efficiency and robustness of our approach concerning the progress in reaching the maximum value of reward. This increase of reward development explains

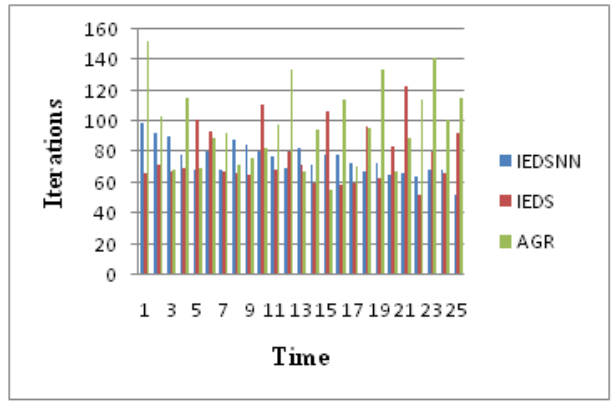


Figure 3. Average capturing time after 30 pursuits

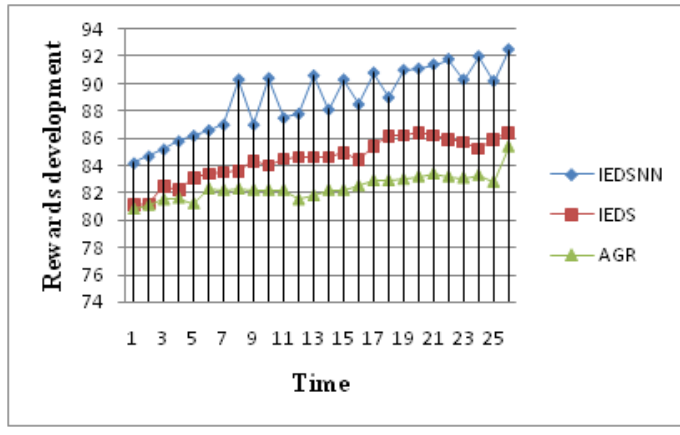


Figure 4. The pursuers' rewards development

clearly the training procedure of the pursuers in forming the appropriate coalitions attempt to make the shortest approximate time in capturing time.

The stability degree of the changing role of pursuers is shown in Figure 6, when we denote that the pursuers stop changing roles for the first 30 iterations and this is a well-proof of well learning process in our CNN.

To enrich our study, we use a static evader to understand the impact of the type of evader on the dynamics of the model. In this case, the evader only moves during the first iteration and stills static among the whole pursuit game process.

Figure 7 reflects the changes regarding the average capturing time using static evaders in comparison with the dynamic evaders used in the previous experiments. The significant advantage of the proposed method is the fast learning concern-

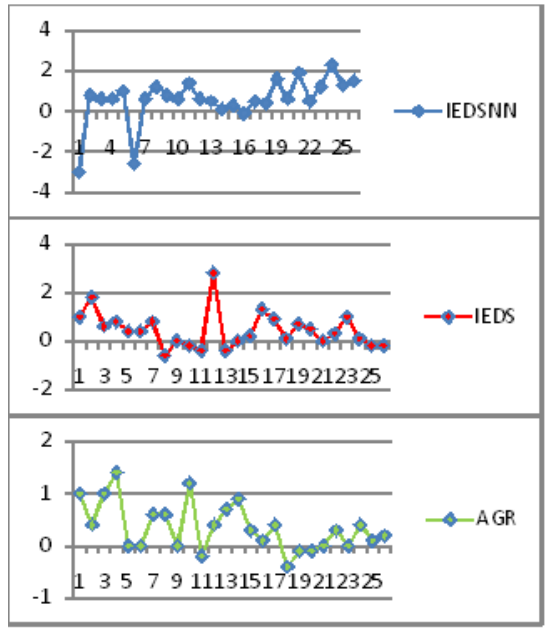


Figure 5. Average pursuers' reward per iteration

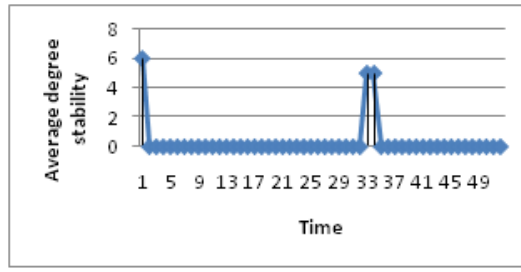


Figure 6. Average degree stability

ing the formation of an optimal coalition making a speed capture than depending on dynamic evader. The results prove that the type of evader affects the overall learning behavior of the simulation. More broadly, these results indicate that IEDSNN is a promising approach for well prediction that makes a speed capture.

From Figures 8 and 9, it can be seen that there is a proportional relationship between the number of iterations performed and the reward obtained for each iteration. When the system starts to well learn the process of chasing, it makes

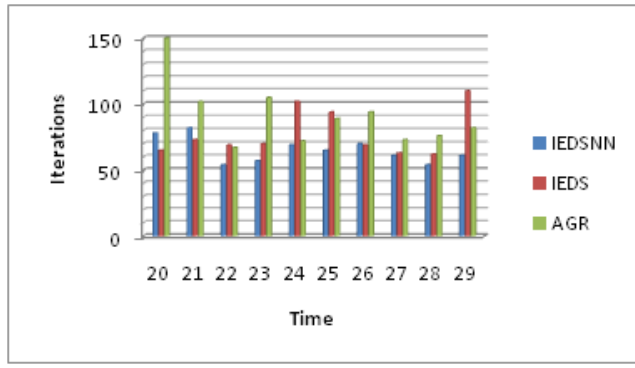


Figure 7. Average capturing time after 30 pursuits

a decrease of 17 in average capturing time which leads to the increase in average reward obtained in a given iteration.

Figure 9 shows the adequate dynamism degree of the roles' changes provided by the new proposal. This reduction, in comparison with the previous model using dynamic evaders, reflects the efficiency of CNN with static evaders. These results confirm the influence of the type of agent in PEG.

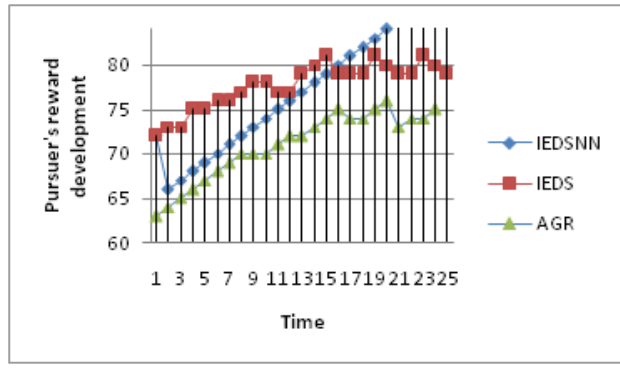


Figure 8. The pursuers' rewards development

7 CONCLUSION

In this paper, we have proposed a new pursuit coalition formation algorithm based on IEDS techniques as well as convolutional neural networks to allow the dynamic grouping of the implied pursuers. The used principles aim to provide appropriate coalition in accordance with the temporal constraints by accelerating the capture

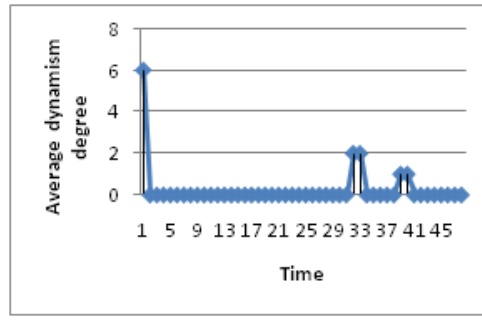


Figure 9. The average dynamism degree

process the detected evaders. From the used learning process, we can easily constate that the pursuers discover an approximated coalition in which they are able to be adapted to its changes after several experiments characterized by different pursuers' behaviors. This fact proves that IEDSNN exploits the principles of both convolutional neural networks and IEDS technique in the pursuit MAS game. To emphasize usefulness of our approach with, we effectuated a comparison study with a decentralized strategy of coalition based on AGR organizational model as well as IEDS algorithm. From the experimental results, we can deduce that this approach improves the pursuit capturing time, the coalition stability, as well as the payoff acquiring performed by the pursuers during the pursuit in comparison with the recent proposed approaches.

A solid foundation in this research is laid which can extend interesting other views in this kind of pursuit-evasion multi-agent game. For future work, we plan to use the learned representations (i.e. average capturing time, average rewards obtained, degree of stability), through our best CNN prediction system, to study the provided performance in different pursuit cases such as agent speeds and environment type. We will propose to use this information at the time of learning in order to learn all the possible coalition formations allowing a quick capture. We will also study the influence of the type of agent and the environment kind on the quality of performance prediction systems.

REFERENCES

[1] Zhou, Z.—Xu, H.: Mean Field Game and Decentralized Intelligent Adaptive Pursuit Evasion Strategy for Massive Multi-Agent System Under Uncertain Environment. 2020 American Control Conference (ACC), IEEE, 2020, pp. 5382–5387, doi: 10.23919/ACC45564.2020.9147659.

[2] Zhang, L.—Li, J.—Zhu, Y.—Shi, H.—Hwang, K. S.: Multi-Agent Reinforcement Learning by the Actor-Critic Model with an Attention Interface. Neurocomputing, Vol. 471, 2022, pp. 275–284, doi: 10.1016/j.neucom.2021.06.049.

[3] Zhou, Z.—Xu, H.: Decentralized Optimal Large Scale Multi-Player Pursuit-Evasion Strategies: A Mean Field Game Approach with Reinforcement Learning. Neurocomputing, Vol. 484, 2022, pp. 46–58, doi: 10.1016/j.neucom.2021.01.141.

[4] Li, Z. Y.—Zhu, H.—Luo, Y. Z.: An Escape Strategy in Orbital Pursuit-Evasion Games with Incomplete Information. Science China Technological Sciences, Vol. 64, 2021, No. 3, pp. 559–570, doi: 10.1007/s11431-020-1662-0.

[5] Wang, Y. F.: A Pursuit Evasion Game Approach to Obstacle Avoidance. Master Thesis. University of Waterloo, 2021.

[6] Tan, X.—Zhou, L.—Wang, H.—Sun, Y.—Zhao, H.—Seet, B. C.—Wei, J.—Leung, V. C. M.: Cooperative Multi-Agent Reinforcement-Learning-Based Distributed Dynamic Spectrum Access in Cognitive Radio Networks. IEEE Internet of Things Journal, Vol. 9, 2022, No. 19, pp. 19477–19488, doi: 10.1109/JIOT.2022.3168296.

[7] Xu, Y.—Yang, H.—Jiang, B.—Polycarpou, M. M.: Multiplayer Pursuit-Evasion Differential Games with Malicious Pursuers. IEEE Transactions on Automatic Control, Vol. 67, 2022, No. 9, pp. 4939–4946, doi: 10.1109/TAC.2022.3168430.

[8] Chipade, V. S.: Collaborative Task Allocation and Motion Planning for Multi-Agent Systems in the Presence of Adversaries. Ph.D. Thesis. University of Michigan, 2022.

[9] Li, Z.—Zhu, H.—Yang, Z.—Luo, Y.: A Dimension-Reduction Solution of Free-Time Differential Games for Spacecraft Pursuit-Evasion. Acta Astronautica, Vol. 163, Part B, 2019, pp. 201–210, doi: 10.1016/j.actaastro.2019.01.011.

[10] Wang, W.—Li, P.: Planning and Formulations in Pursuit-Evasion: Keep-Away Games and Their Strategies. CoRR, 2022, doi: 10.48550/arXiv.2206.08318.

[11] Bravo, L.—Ruiz, U.—Murrieta-Cid, R.: A Pursuit-Evasion Game Between Two Identical Differential Drive Robots. Journal of the Franklin Institute, Vol. 357, 2020, No. 10, pp. 5773–5808, doi: 10.1016/j.jfranklin.2020.03.009.

[12] Liu, M.—Zhang, J.—Shang, M.: Real-Time Cooperative Kinematic Control for Multiple Robots in Distributed Scenarios with Dynamic Neural Networks. Neurocomputing, Vol. 491, 2022, pp. 621–632, doi: 10.1016/j.neucom.2021.12.038.

[13] Souidi, M. E. H.—Piao, S.: A New Decentralized Approach of Multiagent Cooperative Pursuit Based on the Iterated Elimination of Dominated Strategies Model. Mathematical Problems in Engineering, Vol. 2016, 2016, Art. No. 5192423, doi: 10.1155/2016/5192423.

[14] Khosravifar, B.—Bouchet, F.—Feyzi-Behnagh, R.—Azevedo, R.—Harley, J. M.: Using Intelligent Multi-Agent Systems to Model and Foster Self-Regulated Learning: A Theoretically-Based Approach Using Markov Decision Process. 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA), 2013, pp. 413–420, doi: 10.1109/AINA.2013.70.

[15] Castaneda, G.—Morris, P.—Khoshgoftaar, T. M.: Evaluation of Maxout Activations in Deep Learning Across Several Big Data Domains. Journal of Big Data, Vol. 6, 2019, Art. No. 72, doi: 10.1186/s40537-019-0233-0.

[16] Cai, M.—Liu, J.: Maxout Neurons for Deep Convolutional and LSTM Neural Networks in Speech Recognition. Speech Communication, Vol. 77, 2016, pp. 53–64, doi: 10.1016/j.specom.2015.12.003.

[17] Srivastava, K.—Surana, A.: Multi Agent AI for Tactical Maneuvering. In: Pham, T., Solomon, L. (Eds.): Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications IV. Proceedings of the SPIE, Vol. 12113, 2022, pp. 166–176, doi: 10.1117/12.2617157.

[18] Raboin, E.: Model-Predictive Strategy Generation for Multi-Agent Pursuit-Evasion Games. Ph.D. Thesis. University of Maryland, College Park, 2015.

[19] Yan, T.—Cai, Y.—Xu, B.: Evasion Guidance Algorithms for Air-Breathing Hypersonic Vehicles in Three-Player Pursuit-Evasion Games. Chinese Journal of Aeronautics, Vol. 33, 2020, No. 12, pp. 3423–3436, doi: 10.1016/j.cja.2020.03.026.

[20] Ruiz López, U.: Pursuit-Evasion Problems with a Differential Drive Robot and an Omnidirectional Agent. Ph.D. Thesis. CIMAT, Guanajuato, Mexico, 2020.

[21] Hespanha, J. P.—Prandini, M.—Sastry, S.: Probabilistic Pursuit-Evasion Games: A One-Step Nash Approach. Proceedings of the 39th IEEE Conference on Decision and Control, Vol. 3, 2000, pp. 2272–2277, doi: 10.1109/CDC.2000.914136.

[22] Guo, M.—Xin, B.—Chen, J.—Wang, Y.: Multi-Agent Coalition Formation by an Efficient Genetic Algorithm with Heuristic Initialization and Repair Strategy. Swarm and Evolutionary Computation, Vol. 55, 2020, Art. No. 100686, doi: 10.1016/j.swevo.2020.100686.

[23] Krausburg, T.: Hierarchical Coalition Formation in Multi-Agent Systems. In: Rodríguez González, S., González-Briones, A., Gola, A., Katranas, G., Ricca, M., Loukanova, R., Prieto, J. (Eds.): Distributed Computing and Artificial Intelligence, Special Sessions, 17th International Conference (DCAI 2020). Springer, Cham, Advances in Intelligent Systems and Computing, Vol. 1242, 2021, pp. 210–214, doi: 10.1007/978-3-030-53829-3_23.

[24] Estrada, R.—Mizouni, R.—Otrok, H.—Mourad, A.: Task Coalition Formation for Mobile CrowdSensing Based on Workers' Routes Preferences. Vehicular Communications, Vol. 31, 2021, Art. No. 100376, doi: 10.1016/j.vehcom.2021.100376.

[25] Babu, T. S. K.—Chitnis, S.: Coalition Formation Based Cooperation Strategy for Routing in Delay Tolerant Networks. Materials Today: Proceedings, Vol. 45, Part 9, 2021, pp. 8182–8187, doi: 10.1016/j.matpr.2021.02.694.

[26] Souidi, M.—Piao, S.—Li, G.—Chang, L.: Coalition Formation Algorithm Based on Organization and Markov Decision Process for Multi-Player Pursuit Evasion. Multiagent and Grid Systems, Vol. 11, 2015, No. 1, pp. 1–13, doi: 10.3233/MGS-150226.

[27] Cruz, D. L.—Yu, W.: Path Planning of Multi-Agent Systems in Unknown Environment with Neural Kernel Smoothing and Reinforcement Learning. Neurocomputing, Vol. 233, 2017, pp. 34–42, doi: 10.1016/j.neucom.2016.08.108.

[28] Souidi, M. E. H.—Songhao, P.—Guo, L.—Lin, C.: Multi-Agent Cooperation Pursuit Based on an Extension of AALAADIN Organisational Model. Journal of Experimental and Theoretical Artificial Intelligence, Vol. 28, 2016, No. 6, pp. 1075–1088, doi: 10.1080/0952813X.2015.1056241.

[29] Benoudina, L.—Redjimi, M.: Multi Agent System Based Approach for Industrial Process Simulation. Journal Européen Des Systèmes Automatisés (JESA), Vol. 54, 2021, pp. 209–217, doi: 10.18280/jesa.540202.

[30] Boudjidj, A.—Merah, E.—Souidi, M. E. H.: Towards a Formal Multi-Agent

Organizational Modeling Framework Based on Category Theory. Informatica, Vol. 45, 2021, No. 2, doi: 10.31449/inf.v45i2.2967.

[31] Qadir, M. Z.—Piao, S.—Jiang, H.—Souidi, M. E. H.: A Novel Approach for Multi-Agent Cooperative Pursuit to Capture Grouped Evaders. The Journal of Supercomputing, Vol. 76, 2020, No. 5, pp. 3416–3426, doi: 10.1007/s11227-018-2591-3.

[32] Redjimi, K.—Redjimi, M.: A Multi-Agent System for Industrial Simulators Design. In: Troiano, L., Vaccaro, A., Tagliaferri, R., Kesswani, N., Díaz Rodriguez, I., Brigui, I., Parente, D. (Eds.): Advances in Deep Learning, Artificial Intelligence and Robotics (icdlair 2020). Springer, Cham, Lecture Notes in Networks and Systems, Vol. 249, 2022, pp. 129–140, doi: 10.1007/978-3-030-85365-5_13.

[33] Liu, X.—Masoud, N.—Zhu, Q.—Khojandi, A.: A Markov Decision Process Framework to Incorporate Network-Level Data in Motion Planning for Connected and Automated Vehicles. Transportation Research Part C: Emerging Technologies, Vol. 136, 2022, Art. No. 103550, doi: 10.1016/j.trc.2021.103550.

[34] Rocholl, N.: Isolating Wildfires Using a Convolutional Neural Network Based Multi-Agent System. Master Thesis. University of Groningen, 2021 (Bachelor Thesis).

[35] Iranfar, A.—Zapater, M.—Atienza, D.: Multiagent Reinforcement Learning for Hyperparameter Optimization of Convolutional Neural Networks. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 41, 2022, No. 4, pp. 1034–1047, doi: 10.1109/TCAD.2021.3077193.

Nabila Sid is a Ph.D. student in the Department of Mathematics and Computer Science, at Khenchela University, Algeria, a member of ICOSI Laboratory. Her research interests include multi-agent system, game theory and convolutional neural network.

Meriem Djezzar is an Associate Professor in computer science at the University of Khenchela, Algeria and a member of the ICOSI Laboratory. She completed her Ph.D. in computer sciences from University of Constantine 2 in 2016. Her research interests are in machine learning, cyber-physical systems, knowledge representation and reasoning.

Mohammed El Habib Souidi received his B.Sc. degree in computer science from the University of Khenchela, Algeria in 2011. He also received his Master's degree in computer science from the same university in 2013, and his Ph.D. in computer science from the Harbin Institute of Technology, China, in 2017. He is working as a Lecturer in the Department of Mathematics and Computer Science in University of Khenchela, Algeria. Moreover, he is affiliated as a researcher in ICOSI Lab, Unniversity of Khenchela. His research interests include multiagent task coordination, reinforcement learning, game theory, and path planning.

Mounir Hemam is an Associate Professor at the University of Khenchela, Algeria and the Head of the GECO Research Group at the ICOSI Laboratory. He has completed his Ph.D. degree in computer sciences from the Mentouri University in January 2012. He received his habilitation qualifications (Accreditation to supervise research HDR) in 2017. His research interests include knowledge representation and reasoning; formal knowledge representation, machine learning and semantic IoT interoperability in the context of Industry 4.0.

Computing and Informatics, Vol. 42, 2023, 568–590, doi: 10.31577/cai_2023_3_568

MODEL FOR SPATIOTEMPORAL CRIME PREDICTION WITH IMPROVED DEEP LEARNING

Ature Angbera

Universiti Sains Malaysia
School of Computer Sciences
Pulau Pinang 11800, Malaysia
&
Department of Computer Science
Joseph Sarwuan Tarkaa University Makurdi
Benue State, Nigeria
e-mail: angberaature@student.usm.my

Huah Yong Chan

Universiti Sains Malaysia
School of Computer Sciences
Pulau Pinang 11800, Malaysia
e-mail: hychan@usm.my

Abstract. Crime is hard to anticipate since it occurs at random and can occur anywhere at any moment, making it a difficult issue for any society to address. By analyzing and comparing eight known prediction models: Naive Bayes, Stacking, Random Forest, Lazy:IBK, Bagging, Support Vector Machine, Convolutional Neural Network, and Locally Weighted Learning – this study proposed an improved deep learning crime prediction model using convolutional neural networks and the xgboost algorithm to predict crime. The major goal of this research is to provide an improved crime prediction model based on previous criminal records. Using the Boston crime dataset, where our larceny crime dataset was extracted, exploratory data analysis (EDA) is used to uncover patterns and explain trends in crimes. The performance of the proposed model on the basis of accuracy, recall, and f-measure was 100 % outperforming the other models used in this study. The analysis of the proposed model and prediction can aid security services in making better use of

their resources, anticipating crime at a certain time, and serving the society better.

Keywords: Crime prediction, deep learning, spatiotemporal, data mining, ensemble learning

1 INTRODUCTION

Human behavior disorder is the leading cause of crimes that wreak havoc on society in many of ways. A crime is a societal illness that affects every sector of the society in a region where it happens. The crime rate is very high in the developing countries. Governments around the globe expend a lot of resources trying to deal with crime, but since crime is very complex in its nature [1], it is always very difficult to tackle it manually in traditional ways. The information communication technology (ICT) can efficiently help dealing with this problem.

Like a disease, crime is a society issue that tends to proliferate in geographic clusters. Since crime is a geographic phenomenom its hotspots, spatial clusters, spatial correlations of various indicators and forecasts provide the common topics for the crime research [2]. Spatiotemporal crime prediction with the latest artificial intelligent techniques is very important. And for public safety and smart city operations spatiotemporal crime prediction is critical [3]. Because crime episodes are sparsely dispersed spatially and temporally, the traditional deep learning approaches backed only by a coarse location-scale can forecast crime density to a limited extent. Law enforcers require precise data regarding illegal activity in order to foresee, respond and solve spatiotemporal illegal conduct.

Anticipating when and where a crime will occur, often referred to as "predictive policing", permits a society to dispatch law enforcers to highly crime potencial regions or circumstances prior to a crime occurring. Criminal activity can be predicted spatially and temporally which is helpful for a targetable allocation of police resources and surveillance. Advanced deep learning techniques are effective tools for predicting future events based on the behavior of previous ones. However, the exponential growth of spatiotemporal data is only rarely used for anticipating crime events [4] using a repository of spatiotemporal crime data sets. The availability of spatiotemporal crime data has already facilitated the development of data-driven strategies for predicting the occurrence of crimes in recent years [5, 6].

The feature representation efficacy of neural network design distinguishes deep learning-based methods from other spatiotemporal prediction methods. Many recently proposed forecasting frameworks, such as attentional neural methods [7], convolution-based learning approach [8], and spatial relation encoder with graph neural networks [9], Spatiotemporal Sequential Hypergraph Network [10], has been focused on modeling time-evolving regularities over the temporal dimension and

the underlying regional geographical dependencies over the spatial dimension. Despite their success, we believe that conventional spatiotemporal prediction models fall short meeting the particular problems that multi-dimensional crime data [11] presents. There are explicit and implicit relationships between different kinds of crimes because of the heterogeneity of crime data. Current approaches are inherently incapable of capturing cross-type crime influences in a fully dynamic scenario involving both spatial and temporal patterns due to their inherent architecture.

In this study, we therefore proposed an improved deep learning technique for spatiotemporal crime prediction, using deep convolutional neural networks as the feature extractor and a strong ensemble metal classifier known as XGBoost algorithm for final prediction. Another important angle of this study is to show that crime has always been studied from literature in quantitative terms which combines many crimes to be worked on, thus, making the designed systems less productive. Hence in our study, we have studied crime giving a room for more understanding of the crime and better police allocation. Crime is a legally punished conduct, it is detrimental to society, therefore it is necessary to comprehend crime in order to prevent criminal action [12]. The major goal of this paper is to provide an improved crime prediction model based on previous criminal records.

2 RELATED STUDIES

Several approaches in regards to crime prediction have been presented in recent times to provide police officers with efficient and persuasive knowledge for effective resource allocation in order to avoid future crimes [13, 14]. In [15], the article presented a crime prediction model that utilizes hotspot analysis to enhance its accuracy. The model comprises three phases: Crime Hotspot Identification, Dataset Preparation, and Crime Prediction Approach. In the initial phase, hotspot analysis is employed to pinpoint areas with high crime incidence. In the second stage, the location coordinates are replaced by the cluster number to which they belong, and the modified dataset is used to train the crime prediction model. In the third and final phase, the trained model is utilized to categorize each instance into one of 37 crime categories using advanced techniques like Naive Bayes, Decision Tree, and to ensemble learning approaches. The outcomes of the study demonstrate that incorporating hotspot analysis into the model leads to a significant improvement in crime prediction accuracy. The results indicate that Voting with Naive Bayes and REPTree produce the most reliable classification results, although deep learning could have been utilized for better results. However, the study only uses crime data from one year, which may not be adequate to capture long-term trends or changes in crime patterns. Unlike the study, our research employs a dataset that spans more than one year. In [16], the article introduces a technique for examining the strength and spatiotemporal progression of hotspots identified by the EFCM algorithm [17] for spatiotemporal hotspot detection. The proposed method in the article introduces a novel approach for analyzing the spatiotemporal evolution of

hot spots in a specific area by calculating the hot spot strength index, which measures the percentage of time a selected area is affected by hot spots. Furthermore, the method can assess the reliability of the evaluation by calculating a reliability index based on a hot spot reliability measure proposed in the previous study. The application of this method in crime analysis of the City of London using a dataset of criminal events since 2011 shows a decrease in the frequency of all types of criminal events across the study area in the recent years. While the study lacks a detailed comparison of the proposed method with other existing machine learning methods, our study presents a comparison with state-of-the-art machine learning models to fully evaluate the effectiveness of our proposed method. A genetic-fuzzy system was created in [18] to produce an intelligible fuzzy knowledge base that includes patterns for forecasting future spatiotemporal crimes. The system consists of three steps: fuzzy problem space partitioning, meaningful feature selection, and fuzzy knowledge base construction. A generated dataset and a real-world dataset from Tehran, Iran were used to test the suggested system. The results suggest that the proposed approach is a good tool for detecting patterns and forecasting future crimes in contexts where crimes are concentrated in the location and timeaspect. The authors of the article reported a high computational complexity of the method, but they had no specified the extent of it. However, in our study, we proposed an improved deep learning model that can reduce the computational cost. In [19], the article introduces a novel deep learning technique called Geographic-Semantic Ensemble Neural Network (GSEN), which stacks a geographic prediction neural network and a semantic prediction neural network to improve prediction accuracy. The GSEN model combines various structures, including Predictive Recurrent Neural Network (PredRNN), Graph Convolutional Predictive Recurrent Neural Network (GC-PredRNN), and Ensemble Layer, to capture spatiotemporal dynamics from different perspectives. The RMSE of the suggested system was 0.6425 ± 0.0057. However, an improved deep learning model is presented in our study which has lower values for RMSE. In [20], an XGBoost classifier was developed for determining if a seven-day sliding time frame within a given county contains or does not contain a human trafficking-related incident. A case study was conducted with a new combined human trafficking criminal dataset that had a Matthews correlation value of 0.86. However, better advanced deep learning models would have been used for a better result. In [21], a deep learning-based model for spatiotemporal crime prediction using convolutional neural networks is proposed. The proposed approach uses a hierarchical structure to understand the timing of criminal events, with branches that focus on different time periods. Additionally, it utilizes a channel projection to better understand how past events may impact future crime risk. The effectiveness of this model is assessed using publicly available crime data sets from Chicago and Los Angeles, and compared to traditional methods. The proposed model (CNN-PT) outperforms the traditional models in terms of both AP score and RMSE score. The temporal hierarchical structure of the proposed model improves the AP performance of traditional CNN models by 1.4 % in the Chicago dataset and 1.7 % in the Los Angeles dataset. Additionally, the channel projection further

improves the AP performance by 0.6 % in the Chicago dataset and 0.7 % in the Los Angeles dataset. The authors of [22], suggested a system called Crime Situation Awareness Network (CSAN) to predict future crime situations by utilizing multi-correlations and sequential context information. To achieve this, they developed a new neural structure consisting of a Conv-VAE information compression component and a Context-based Sequence Generative Model temporal component. Data preparation included creating detailed Crime Situation Awareness Graphs (CSAGs) and conducting statistical analysis. The performance of the CSAN was measured using metrics like RMSE, MSPE, and JS. In order to predict a person's likelihood of committing a crime and the type of crime they are most likely to commit based on their criminal charge history data, Chun et al. used deep neural networks (DNNs) as a machine-learning technique [23]. "Deep inception-residual networks (DIRNet)" were proposed by Ye et al. (2021) to forecast fine-grained theft-related crimes using a non-emergency service request data (311 events). The method involves identifying low-level spatiotemporal correlations from crime events and complaint records in the 311 dataset using inception units made up of asymmetrical convolution layers. Data from New York City's 311 system and theft-related offences from 2010 to 2015 are used to assess DIRNet's performance. The findings indicate that DIRNet achieves an average F1 score of 71 %, which outperforms other prediction models [3]. However, improved deep learning will produce better results for efficient policing.

The review discusses several approaches to crime prediction, including a crime prediction model that uses hotspot analysis to improve accuracy. The model has three phases: Crime Hotspot Identification, Dataset Preparation, and Crime Prediction Approach. The results show that the proposed model significantly improves the accuracy of crime prediction. Other approaches discussed in the review include an Extended Fuzzy C-means (EFCM) spatiotemporal hot spot detection algorithm, a genetic-fuzzy system, a Geographic-Semantic Ensemble Neural Network (GSEN), and an XGBoost classifier. Each method has its own advantages and disadvantages. The effectiveness of these models is assessed using different evaluation metrics, and compared to traditional methods which other studies failed to do. Hence, our study proposes an improved deep learning model which further improve the accury of the prediction model compared with the traditional methods. Finally, the researchers came to the conclusion that by incorporating dynamic variables over a wide range of criminal occurrences and with the high growth of spatiotemporal crime datasets, crime prediction performance might be greatly enhanced for better policing.

3 PROPOSED MODEL

We have covered the suggested spatiotemporal crime prediction approach in this section. The "Convolutional Neural Network (CNN)" and the "Extreme Gradient Boosting (XGBoost)" classifier, which are the components needed to make predictions in the proposed model is presented. The proposed model is depicted in Figure 1. In this study, Deep Convolutional Extreme Gradient Boosting (DeCXG-

Boost) model, which combines these two models, is proposed and will be used to predict crime spatiotemporally.

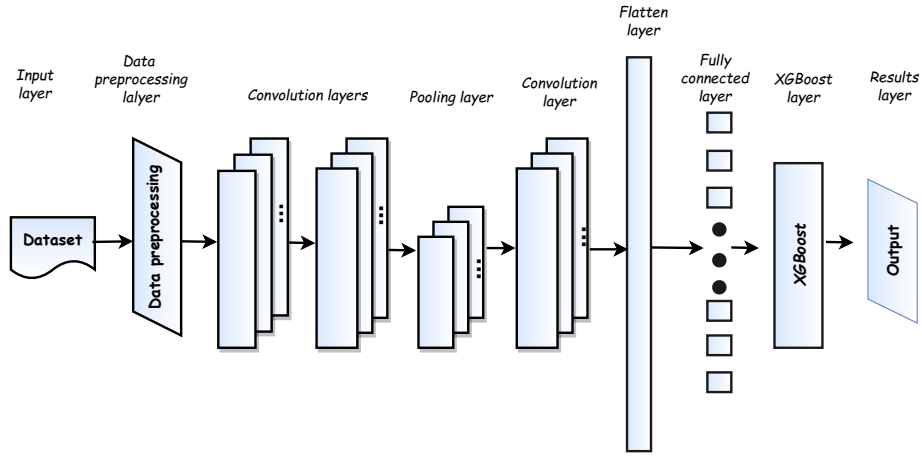


Figure 1. The proposed frame work

3.1 Convolutional Neural Networks (CNNs)

A deep, feed-forward neural network is known as a CNN [24] that is frequently used to analyse visual imagery [25]. The traditional form of CNNs is the classic multi-layer perceptron (MLP). Despite the fact that CNNs were not designed expressly for non-image data, they have been widely used in spatiotemporal data-mining applications including trajectory and spatiotemporal raster data [26]. Figure 2 depicts the architecture of a CNNs.

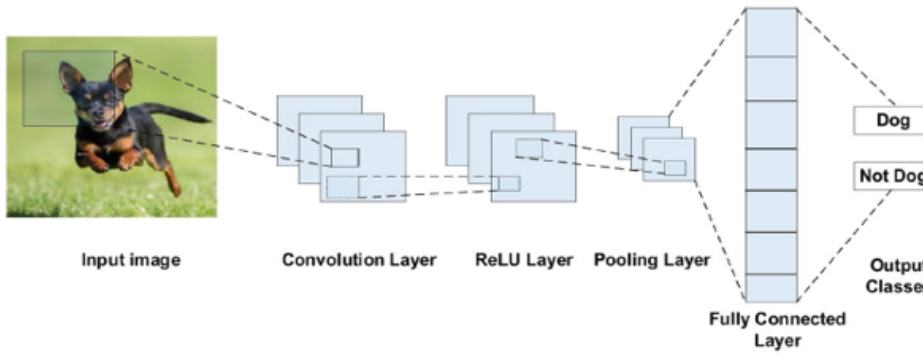


Figure 2. Architecture of CNN [27]

The feature outcome map is created by convolving a one-dimensional entry $x = (x_t)_{N-1}^{t=0}$ of size N in the first layer with a set of M_1 3-dimensional filters, w_h^1 for $h = 1; \ldots; M_1$, for which the filters are applied to all input channels [28]: see Equation (1).

$$a^1(i,h) = \left(w_h^1 \times x\right)(i) = \sum_{j=\infty}^{\infty} w_h^1(j) \times (i-j), \tag{1}$$

where $w_h^1 \in R^{1\times k\times 1}$ and $a^1 \in R^{1\times N-1+1\times M_1}$.

There'll be a single input pathway, and the first layer's output will be routed via the non-linear activation function $h(\cdot)$ to produce $f^1 = h(a^1)$.

A convolutional layer, a pooling layer, and a fully connected layer make up the hidden layer. Using learnable filters, the convolutional layer harvests information from various parts of the raw input or intermediate feature maps autonomously [29]. The pooling layer adds all of the items in the pooling frame together. This approach uses a max-pooling operation to reduce the dimensionality of the input tier by selecting the highest value from each subregion of the preceding layer [29]. Consequently, this level lowers the learning process's computing cost and handles any overfitting difficulties [30].

In [31], shows that, the hidden layer $l = 2; \ldots; L$, the input feature map $f^{l-1} \in R^{1\times N_{l-1}\times M_{l-1}}$, where $1 \times N_{l-1} \times M_{l-1}$ is the size of the output filter map from the previous convolution with $N_{l-1} = N_{l-2} - k + 1$, is convolved with a set of M_1 filters $w_h^1 \in R^{1\times k\times M_{l-1}}$, $h = 1; \ldots; M_1$, to create a feature map $a^1 \in R^{1\times N_l\times M_l}$ as follows in Equation (2) [28].

$$a^1(i,h) = \left(w_h^l \times f^{l-1}\right)(i) = \sum_{j=\infty}^{\infty} \sum_{m=1}^{m_{l-1}} w_h^l(j,m) f^{l-1}(i-j,m). \tag{2}$$

To create the expected output, the fully connected layer flattens and incorporates the high-level obtained attributes learnt by the convolution layer. The attributes figures are then put into $f^1 = h(a^1)$ using non-linear activation functions.

After L convolutional layers, the network produces the matrix f^L, whose size is determined by the filter size and figure of filters employed in the last layer [28]. In a nutshell, the full connected layers acquire the mid and low-level characteristics and generate the high-level abstraction, that represents the final-level layers, just like in a traditional neural network. The classification scores are provided by the last layer (example SVM, etc.). Every score represents the likelihood of a particular class in a given situation [27]. In our study we chose the xgboost classifier on the last-stage layer.

3.2 XGBoost Classification Algorithm

The XGBoost discussed in [32] was created using a GBDT (Gradient Boosting Decision Tree), and it was shown to have excellent convergence and generalisation

speed [33]. In [33], the XGBoost algorithm's goal function and optimization strategy were introduced. XGBoost's target function is given by Equation (3) [34].

$$Obj(\theta) = L(\theta) + \Omega(\theta), \tag{3}$$

where $L(\theta) = l(y_i', y_i)$ and $\Omega(\theta) = \gamma T + \frac{1}{2}\lambda||\omega||^2$.

The objective function is divided into 2 sections: $L(\theta)$ and $\Omega(\theta)$, which correspond to the formula's numerous parameters. The difference between the forecast y_i and the target y_i is measured by $L(\theta)$, a differentiable convex loss function. The point is to demonstrate how we can incorporate the facts into the framework [34]. Convex loss functions that are frequently employed, such as the mean square loss function in Equation (4) and the Logistic loss function shown in Equation (5), can be employed in the following equation.

$$l(y_i', y_i) = (y_i' - y_i)^2, \tag{4}$$

$$l(y_i', y_i) = y_i \ln\left(1 + e^{-y_i'}\right) + (1 + y_i) \ln\left(1 + e^{y_i'}\right). \tag{5}$$

Complex models are penalised by the regularised term $\Omega(\theta)$. T is the number of leaves in the tree, and y is the learning rate, which ranges from 0 to 1. When multiplied by T, it equals spanning tree pruning, which prevents overfitting. When compared to the classic GBDT algorithm, the XGBoost algorithm increases the term $\frac{1}{2}\lambda||\omega||^2$. The regularized parameter is λ, while w is the weight of the leaves. This item's value can be increased to control the model from fitting and to improve its generalisation capabilities. The inclusion of model penalty items with functions as parameters, on the other hand, leads in the failure of classical approaches to be optimised by the objective function in Equation (3). As a result, we must assess if we can to learn to obtain the aim y_i as seen in Equation (6) [34]:

$$L(\theta) = \sum_{i=1}^{n} l\left(y_i, y_i^{'t-1} + S_t(T_i)\right) + \Omega(\theta), \tag{6}$$

where, in the t iteration, $S_t(T_i)$ denotes the tree produced by instance i.

The optimization target in each iteration is to build a tree design that minimises the aimed function. Hence, when solving square loss function, the objective function of Equation (6) is optimal, but it becomes quite difficult when calculating other loss functions. As a result, Equation (6) translates Equation (7) using the two-order Taylor expansion, allowing further loss functions to be solved.

$$L(\theta) = \sum_{i=1}^{n} \left[l\left(y_i, y_i^{'t-1} + g_i S_t(T_i)\right) + \frac{1}{2} h_i S_t^2(T_i) \right] + \Omega(\theta), \tag{7}$$

where, $g_i = \partial_{(y')}^{t-1} l\left(y_i, y^{'t-1}\right)$ which is the 1st derivative of the error function and $h_i = \partial_y^{'(t-1)2} l\left(y_i, y^{'t-1}\right)$ is the 2nd derivative of the error function.

Because tree model needs to find the best segmentation points and then store them in a number of blocks, the algorithm ranks the eigenvalues based on the realisation of XGBoost. This structure is reused in subsequent iterations, resulting in a significant reduction in computing complexity. Furthermore, the information gain of each feature must be determined during the node splitting process, which employs the greed algorithm as shown in Algorithm 1, allowing the calculation of information gain to be parallelized [33].

Algorithm 1 Split finding greed algorithm

Require: Input I, current node's instance set
Ensure: Input d, dimension of the characteristic

```
gain ← 0
G ← ∑_i ∈ I_{g_i}
for k = 1 to m do
    G_L ← 0
    for j in sorted(I, by X_j k) do
        G_L ← G_L + g_j
    end for
end for
G_L ← G_L + g_j
G_R ← G − G_L
Result:  Split with max score
```

In view of the above overviews of the CNN and XGBoost models, we therefore proposed an improved deep learning model for spatiotemporal crime prediction called DeCXGBoost. The DeCXGBoost model combines two machine learning algorithms, Convolutional Neural Network (CNN) and eXtreme Gradient Boosting (XGBoost), to improve the accuracy of prediction tasks. The CNN algorithm is used to extract high-level features from raw input data, such as images or time-series data. It involves several convolutional layers that perform operations on the input data to extract features and a pooling layer that reduces the dimensionality of the output. The output from the convolutional and pooling layers is then fed into a fully connected layer, which performs classification or regression. The XGBoost algorithm is a gradient boosting framework that is used for supervised learning problems. It builds a series of decision trees iteratively, with each new tree correcting the errors made by the previous one. The DeCXGBoost model combines these two algorithms to leverage their respective strengths. The CNN algorithm is used to extract high-level features from the raw input data, which are then fed into the XGBoost algorithm to make predictions. The combination of these two algorithms allows the model to extract complex features from the input data and make accurate predictions.

Mathematically, the DeCXGBoost model can be represented as follows: CNN: The output of the ith convolutional layer is given by:

$$O_i = f_i(w_i * O(i-1) + b_i), \tag{8}$$

where w_i is the ith set of convolutional filters, $O(i-1)$ is the output of the previous layer, b_i is the bias term, and f_i is the activation function.

XGBoost: The output of the model is given by:

$$Y = F(X) = \sum f_t(X), \tag{9}$$

where X is the input data, f_t is the t^{th} decision tree, and $\sum$ is the sum over all decision trees.

DeCXGBoost: The DeCXGBoost model combines the two algorithms by using the output of the final fully connected layer in the CNN as input to the XGBoost algorithm:

$$Y^* = F(X) = \sum f_t(O_L), \tag{10}$$

where O_L is the output of the final fully connected layer in the CNN, and f_t is the t^{th} decision tree in the XGBoost algorithm. Overall, the DeCXGBoost model is a powerful machine learning algorithm that can be used for a wide range of prediction tasks, particularly in areas that involve complex input data such as images and time-series data.

3.3 Dataset

The Boston Police Department's (BPD) criminal event records were employed in this study that documented the incidents to which BPD officers respond. This was a collection of data from the new crime incident reports, which was designed to capture the sort of incidents as well as when and where it occurred. Table 1 shows the attributes of the crime dataset.

	Incident-Number	Offense-Code	Offense-Desc	...	Street	Lat
0	I182070945	619	Vandalism	...	Lincoln ST	42.35779134
1	I182070915	614	Auto theft	...	Hecla ST	42.30682138
2	I182070893	613	Verbal dispute	...	Dehil ST	42.32701648
...	...	...	...	...	...	...
319071	I030217815-08	1843	Larceny	...	Capen ST	42.28647012
319072	I030217815-08	301	Harassment	...	Lawn ST	42.3256949
319073	I010370257-00	3801	Trespassing	...	Hecla ST	42.31731905
	[319074 × 17]					

Table 1. Features for the crime dataset

This spatiotemporal crime dataset consists of seventeen (17) features (columns) and three hundred nineteen thousand and seventy-four (319074) samples (rows).

There are eight categories (Incident number, offense code group, offense description, district, occurred on date, day of week, UCR part, street) and nine numerical (offense code, reporting area, shooting, year, month, hour, lat, long, location) qualities.

3.4 Data Preprocessing

Data cleansing is a process that must be completed prior to data analysis. It entails tasks including filling in missing data, removing discrepancies, and finding outliers [35]. One of the most significant transformations to perform to data is feature scaling. The numeric features employed in the input should not have different scales for ML algorithms to perform properly [35]. As a result, the min/max normalisation approach was used to rescale the data set so that the values on distinct scales in the data set varied in the range of 0–1. The following formula (see Equation (11)) is used to translate a value that falls within the range of 0 to 1.

$$x_{new} = \frac{x - \min(x)}{\max(x) - \min(x)}. \tag{11}$$

3.5 Modeling

The model was trained and tested for the study, so the dataset was split in half in a 75:25 ratio for the models, 75 % of the dataset was utilized to train the model, while 25 % was used to test it. The process of modeling was carried out using the proposed methodology depicted in Figure 3.

4 EXPERIMENTAL RESULT AND ANALYSIS

In this paper, the Boston Police Department's (BPD) crime dataset is used to extract the features of the larceny crime data which we used in this study for analyzing and predicting crime as a type of crime, which is one of our objectives and motivations for this study. According to [36], there exist eight distinct categories of larceny offenses. Out of the eight categories, six are categorized as "non-occupational" offenses, which involve crimes like shoplifting, theft from a vehicle, theft of vehicle parts, pocket-picking, purse snatching, and theft from a coin-operated device. The two other categories are "theft from a building" and "all other larcenies", which are partially categorized as non-occupational and partially as indeterminate. In our study we categorized them as larceny and larceny from motor vehicle.

4.1 Exploratory Data Analysis (EDA)

A script was ran to investigates numerous distinct categories of larceny offences in the dataset, which we classified into two crime categories as previously described. The distribution of crime is depicted in Figure 4. Larceny is the most common sort of crime, followed by larceny from motor vehicles, as shown in Figure 4 below.

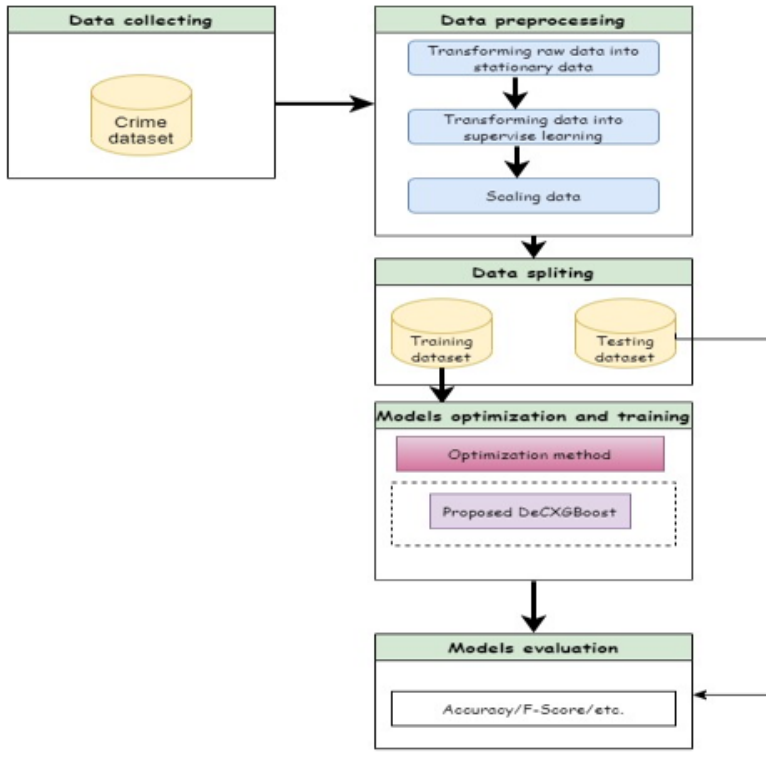


Figure 3. Flow diagram of modeling

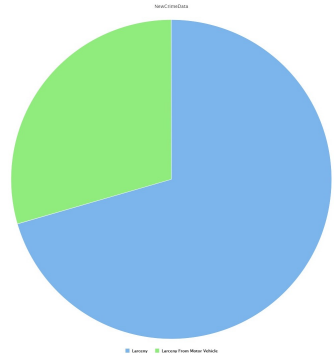


Figure 4. Larceny crime dataset distribution

The extracted crime dataset has 36 782 crime observations, 25 935 of this crimes are committed in shoplifting, pocket picking, and 10847 of this crimes were committed from motor vehicles. Figure 5 shows the hourly committing of these crimes, with larceny out of motor vehicle mostly committed.

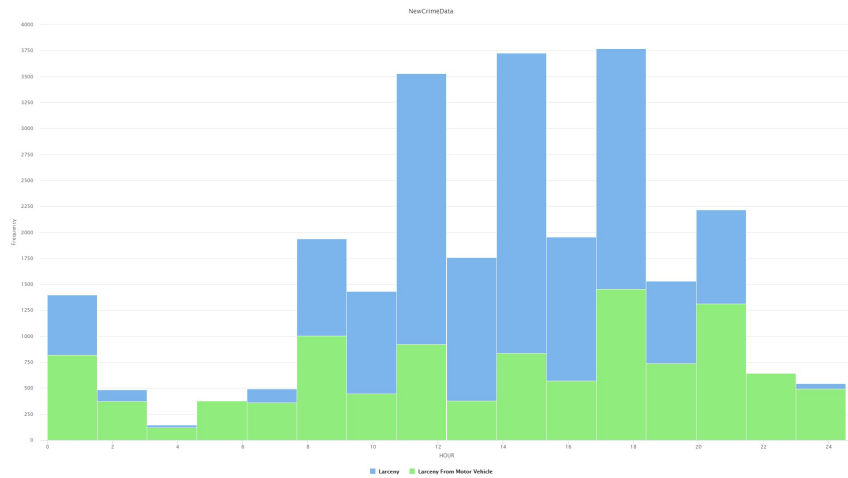


Figure 5. Hourly distribution of crimes committed

Our spatiotemporal crime dataset also displays the locations where the crimes were committed, as seen in Figure 6. It shows that more crimes were committed on Lincoln Street and Lime Street on Wednesdays, Saturdays, and other days. Additionally, Figure 7 shows the districts and the crimes committed on a weekly basis. District D4 has more crimes committed on Fridays, Saturdays, and Wednesdays. With this information and the aid of accurate crime prediction models, security personnel can be more proactive rather than reactive, resulting in a significant reduction in crime within society.

4.2 Prediction Models

The buildup and results of our proposed deep learning improved model is presented, with a comparison to other state-of-the-art models such as Naïve Bayes (NB), Stacking (STK), Random Forest (RF), Lazy:IBK (IBK), Bagging (BAG), Support Vector Machine (SVM), CNNs, and Locally weighted learning (LWL). Our suggested model, as well as the other eight models in this work, were trained and presented with a variety of setting parameters and feature choices. Both time-related and geographic variables are essential, according to the data exploration section, which explains the spatiotemporal interest. All of the models were trained and tested for the study, so the dataset was split in a 75:25 ratio for all of the models. The model was trained on 75 % of the dataset and tested on the remaining 25 %, as previously mentioned. The

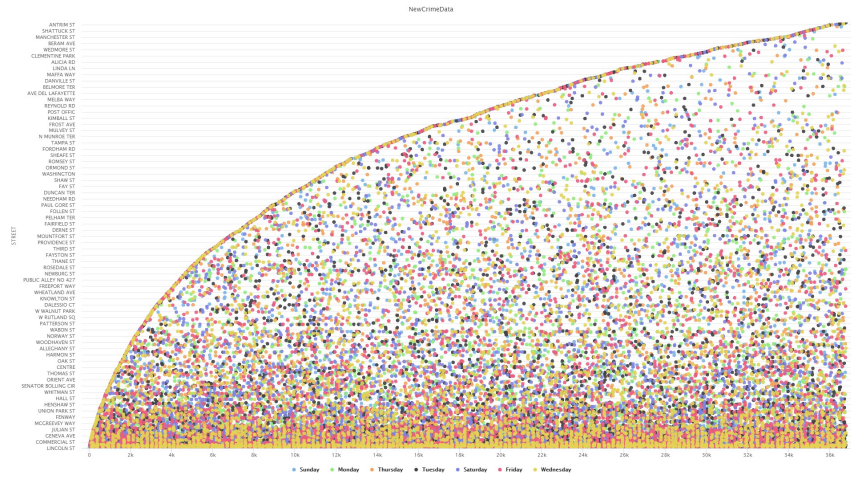


Figure 6. Weekly distribution of crime on streets

CNNs models were constructed using python-3.8, tensorflow-1.01, and keras-1.0 on an Intel core i5 desktop computer. In our proposed model, samples were provided as input. Batch normalization technology and ReLU activation functions were used in all convolution layers. The deep learning model used binary cross-entropy and adaptive moment estimation (Adam) methods as the loss function and optimizer, respectively. Additionally, the He initialization approach was used to initialize the model. WEKA tool was used for the other models. Figure 8 depicts our proposed

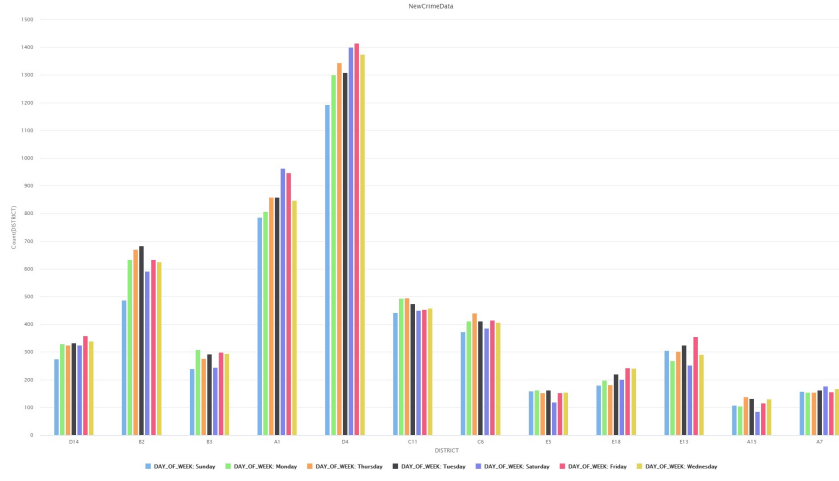


Figure 7. Weekly larceny crimes committed in districts

model's training and testing loss logs, respectively. Figure 9 depicts our suggested model's accuracy log for both training and testing.

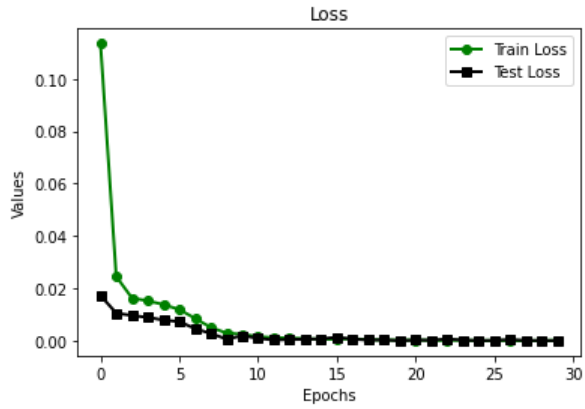


Figure 8. Proposed model loss log

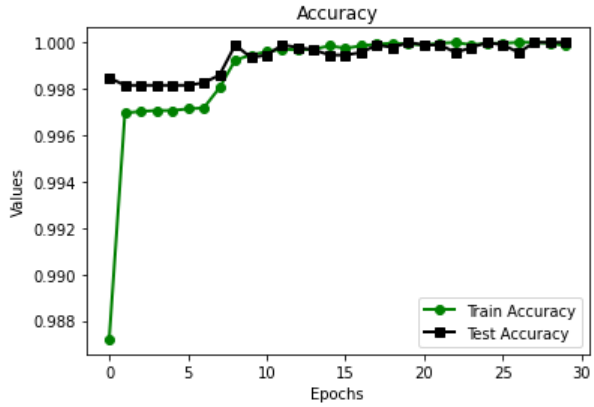


Figure 9. Proposed model accuracy log

The study evaluates the performance of multiple models using various metrics, such as MAE, RMSE, Recall, Accuracy, and F-Measure. The outcomes of these models are shown in Table 2. MAE measures the average absolute deviation between the predicted and actual values, while RMSE is the square root of the average squared deviation between the predicted and actual values. Recall is used to determine the percentage of actual positive cases that were correctly identified by the model. Accuracy is the proportion of correct predictions made by the model, and F-Measure is the harmonic mean of Precision and Recall. The results show that several models

achieved high scores in various performance measures. The NB, RF, BAG, and our proposed model achieved perfect scores in Recall and F-Measure, indicating that they have correctly identified all the positive cases. The CNN model has a relatively low F-Measure score, indicating that it may not perform well in identifying positive cases. The LWL model has the highest RMSE, indicating that it has the largest errors in its predictions. Overall, our proposed model outperformed all the other models, achieving perfect scores in both Recall and F-Measure and the lowest MAE and RMSE scores. The results of this study suggest that the proposed model is highly effective in predicting outcomes in the studied domain.

Models	MAE	RMSE	Recall	Accuracy	F-Measure
NB	0.0005	0.0134	1.000	99.782	1.000
STK	0.4178	0.4592	0.698	69.8097	0.822
RF	0.0629	0.084	1.000	99.891	1.000
IBK	0.0481	0.2192	0.952	95.193	0.952
BAG	0.0001	0.0076	1.000	99.891	1.000
SVM	0.0005	0.0233	0.999	99.7456	0.999
CNN	0.0004	0.0209	1.000	99.565	0.846
OUR's	0.0000	0.0001	1.000	100.000	1.000
LWL	0.913	0.2161	0.946	94.6166	0.945

Table 2. Results of models used in the study

On our well-preprocessed crime dataset with hyperparameter settings and feature selections, Figure 10 illustrates a comparison of the MAE and RMSE of our proposed model and various other models employed in this study. The proposed model is seen to edging out the other models significantly. This is because the lower or small the figure of MAE and RMSE the greater the model.

Figure 11 shows the RECALL and F-Measure of our suggested model vs. the RECALL and F-Measure of other models used in this study. The proposed model appears to greatly outperform the other models. Also Figure 12 compares the prediction accuracy of the proposed model and the other eight models used in this study. The accuracy of our proposed model was higher.

4.3 Discussion

Our proposed (DeCXGBoost) model achieved the best performance across all metrics, with perfect scores in Recall, Accuracy, and F-Measure (see Table 2). The SVM, NB, BAG, and RF models also performed well, achieving high scores in Recall, Accuracy, and F-Measure (see Figures 11 and 12). In contrast, the STK and LWL models had relatively poor performances, with higher MAE and RMSE scores (see Figure 10), and lower Recall and F-Measure scores. The CNN model achieved a high Recall score but had a lower F-Measure score compared to the other models. The use of machine learning algorithms for spatiotemporal crime prediction has been an active area of research in recent years. Various machine

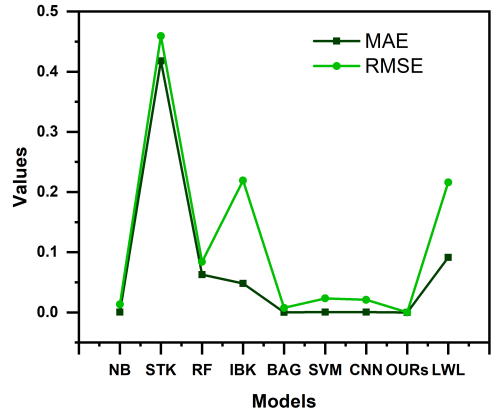


Figure 10. Comparison of our proposed model's performance to the other models using MAE and RMSE

learning algorithms, including deep learning and ensemble methods, have been employed to improve the accuracy of spatiotemporal crime prediction models. One recent study [4] proposed the use of a spatiotemporal convolutional neural network (ST-CNN) to predict crime incidents based on spatiotemporal data. The model used a combination of convolutional neural network and long short-term memory networks to capture both spatial and temporal patterns in crime incidents. The study achieved an accuracy of 86 % in predicting crime incidents. An-

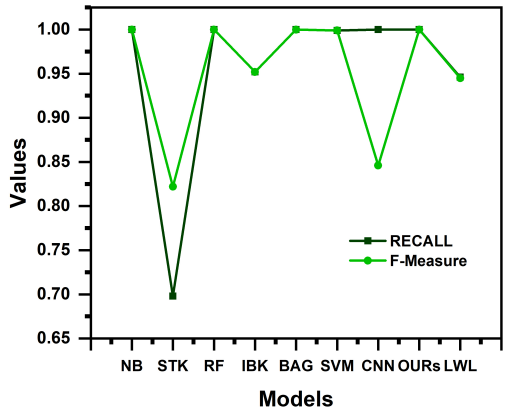


Figure 11. Comparison of our proposed model's performance to the other models using Recall and F-Measure

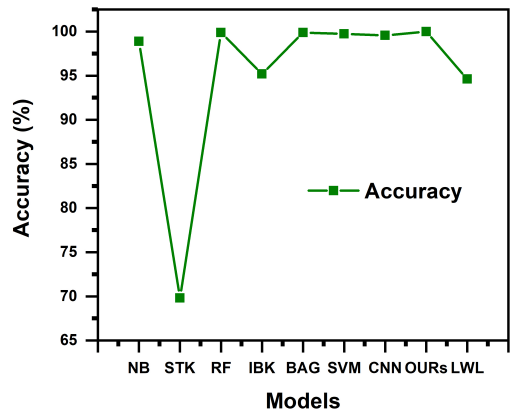


Figure 12. Comparison of our proposed model's performance to the other models using Accuracy

other study [37] used a deep learning approach, specifically the Gated Recurrent Unit (GRU), to predict crime incidents in Los Angeles and Chicago. The model incorporated weather data and social media data in addition to spatiotemporal data to improve its predictions. The study achieved an accuracy of 83.9 % and 86.3 % in predicting crime incidents. Ensemble models, which combine multiple machine learning algorithms, have also been used in spatiotemporal crime prediction. One study [12] proposed an ensemble random forest algorithm to predict crime incidents. The model achieved an accuracy of 99.16 % in predicting crime incidents.

The results of our proposed (DeCXGBoost) model in this study demonstrate the effectiveness of machine learning models in spatiotemporal crime prediction (see Table 2 and Figure 12). The DeCXGBoost model also has the lowest Mean Absolute Error (MAE), and Root Mean Square Error (RMSE) of 0.0000 and 0.0001 respectively making the model the best and more robust when compared to other baseline models used in this study. The study can inform the development of better models and algorithms in the future how to improve the accuracy and efficiency of spatiotemporal crime prediction. Also, the findings can inform the development of better models for prediction tasks in related fields, potentially leading to improvements in various applications, such as healthcare, finance, and cybersecurity.

In summary, this study evaluated the performance of various machine learning models for spatiotemporal crime prediction. The proposed DeCXGBoost model achieved the best performance across all metrics, with perfect scores in Recall, Accuracy, and F-Measure. Other models like SVM, NB, BAG, and RF also performed well. In contrast, the STK and LWL models had relatively poor performances, with higher MAE and RMSE scores and lower Recall and F-Measure scores. The CNN

model achieved a high Recall score but had a lower F-Measure score compared to the other models.

The study also discussed other studies that employed machine learning algorithms for spatiotemporal crime prediction, such as the use of spatiotemporal convolutional neural network (ST-CNN) and the Gated Recurrent Unit (GRU) models. The proposed DeCXGBoost model outperformed these models, achieving a perfect score across all metrics.

The results of this study demonstrate the effectiveness of machine learning models in spatiotemporal crime prediction, and the proposed DeCXGBoost model is highly robust and accurate when compared to other baseline models. The study provides valuable insights into the development of better models and algorithms in the future to improve the accuracy and efficiency of spatiotemporal crime prediction.

5 CONCLUSIONS

This article analyses the outcomes of extracted larceny crime data from the Boston crime dataset and presents exploratory data analysis with a novel proposed spatiotemporal crime prediction model based on classification approaches. Python was used to implement the proposed model. The experimental findings reveal that our suggested DeCXGBoost model outperformed other crime categorization models for all eight methods. For both accuracy and recall, our proposed model received a perfect score. Our proposed methodology can help law enforcement agencies fight crime more effectively, channel resources more efficiently, foresee crime to some extent and serve society. The presented proposed crime prediction model can be used to make predictions and manage resources on any dataset or criminal data. For future improvement, real time crime prediction is an open direction for this work with more advanced technologies.

REFERENCES

[1] ToppiReddy, H. K. R.—Saini, B.—Mahajan, G.: Crime Prediction and Monitoring Framework Based on Spatial Analysis. Procedia Computer Science, Vol. 132, 2018, pp. 696–705, doi: 10.1016/j.procs.2018.05.075.

[2] Matijosaitiene, I.—Zhao, P.—Jaume, S.—Gilkey Jr, J. W.: Prediction of Hourly Effect of Land Use on Crime. ISPRS International Journal of Geo-Information, Vol. 8, 2019, No. 1, Art. No. 16, doi: 10.3390/ijgi8010016.

[3] Ye, X.—Duan, L.—Peng, Q.: Spatiotemporal Prediction of Theft Risk with Deep Inception-Residual Networks. Smart Cities, Vol. 4, 2021, No. 1, pp. 204–216, doi: 10.3390/smartcities4010013.

[4] Esquivel, N.—Nicolis, O.—Peralta, B.—Mateu, J.: Spatio-Temporal Prediction of Baltimore Crime Events Using CLSTM Neural Networks. IEEE Access, Vol. 8, 2020, pp. 209101–209112, doi: 10.1109/ACCESS.2020.3036715.

[5] Wang, H.—Kifer, D.—Graif, C.—Li, Z.: Crime Rate Inference with Big Data. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16), 2016, pp. 635–644, doi: 10.1145/2939672.2939736.

[6] Huang, C.—Zhang, C.—Zhao, J.—Wu, X.—Yin, D.—Chawla, N.: MiST: A Multiview and Multimodal Spatial-Temporal Learning Framework for Citywide Abnormal Event Forecasting. The World Wide Web Conference (WWW '19), 2019, pp. 717–728, doi: 10.1145/3308558.3313730.

[7] Yao, H.—Tang, X.—Wei, H.—Zheng, G.—Li, Z.: Revisiting Spatial-Temporal Similarity: A Deep Learning Framework for Traffic Prediction. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, No. 1, pp. 5668–5675, doi: 10.1609/aaai.v33i01.33015668.

[8] Zhang, J.—Zheng, Y.—Qi, D.: Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 31, 2017, No. 1, doi: 10.1609/aaai.v31i1.10735.

[9] Zhang, X.—Huang, C.—Xu, Y.—Xia, L.—Dai, P.—Bo, L.—Zhang, J.—Zheng, Y.: Traffic Flow Forecasting with Spatial-Temporal Graph Diffusion Network. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, No. 17, pp. 15008–15015, doi: 10.1609/aaai.v35i17.17761.

[10] Xia, L.—Huang, C.—Xu, Y.—Dai, P.—Bo, L.—Zhang, X.—Chen, T.: Spatial-Temporal Sequential Hypergraph Network for Crime Prediction with Dynamic Multiplex Relation Learning. Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21), 2021, pp. 1631–1637, doi: 10.24963/ijcai.2021/225.

[11] Yang, B.—Liu, L.—Lan, M.—Wang, Z.—Zhou, H.—Yu, H.: A Spatio-Temporal Method for Crime Prediction Using Historical Crime Data and Transitional Zones Identified from Nightlight Imagery. International Journal of Geographical Information Science, Vol. 34, 2020, No. 9, pp. 1740–1764, doi: 10.1080/13658816.2020.1737701.

[12] Hossain, S.—Abtahee, A.—Kashem, I.—Hoque, M. M.—Sarker, I. H.: Crime Prediction Using Spatio-Temporal Data. In: Chaubey, N., Parikh, S., Amin, K. (Eds.): Computing Science, Communication and Security (COMS2 2020). Springer, Singapore, Communications in Computer and Information Science, Vol. 1235, 2020, pp. 277–289, doi: 10.1007/978-981-15-6648-6_22.

[13] Mushtaq, H.—Siddique, I.—Malik, B. H.—Ahmed, M.—Butt, U. M.—Ghafoor, R. M. T.—Zubair, H.—Farooq, U.: Educational Data Classification Framework for Community Pedagogical Content Management Using Data Mining. International Journal of Advanced Computer Science and Applications, Vol. 10, 2019, No. 1, pp. 329–338, doi: 10.14569/IJACSA.2019.0100144.

[14] Brayne, S.—Christin, A.: Technologies of Crime Prediction: The Reception of Algorithms in Policing and Criminal Courts. Social Problems, Vol. 68, 2021, No. 3, pp. 608–624, doi: 10.1093/socpro/spaa004.

[15] Hajela, G.—Chawla, M.—Rasool, A.: A Clustering Based Hotspot Identification Approach for Crime Prediction. Procedia Computer Science, Vol. 167, 2020, pp. 1462–1470, doi: 10.1016/j.procs.2020.03.357.

[16] Cardone, B.—Di Martino, F.: Fuzzy-Based Spatiotemporal Hot Spot Intensity and Propagation — An Application in Crime Analysis. Electronics, Vol. 11, 2022, No. 3, Art. No. 370, doi: 10.3390/electronics11030370.

[17] Di Martino, F.—Sessa, S.—Barillari, U. E. S.—Barillari, M. R.: Spatio-Temporal Hotspots and Application on a Disease Analysis Case via GIS. Soft Computing, Vol. 18, 2014, No. 12, pp. 2377–2384, doi: 10.1007/s00500-013-1211-7.

[18] Farjami, Y.—Abdi, K.: A Genetic-Fuzzy Algorithm for Spatio-Temporal Crime Prediction. Journal of Ambient Intelligence and Humanized Computing, 2021, pp. 1–13, doi: 10.1007/s12652-020-02858-3.

[19] Jin, G.—Sha, H.—Feng, Y.—Cheng, Q.—Huang, J.: GSEN: An Ensemble Deep Learning Benchmark Model for Urban Hotspots Spatiotemporal Prediction. Neurocomputing, Vol. 455, 2021, pp. 353–367, doi: 10.1016/j.neucom.2021.05.008.

[20] Ahmed, S.—Gentili, M.—Sierra-Sosa, D.—Elmaghraby, A. S.: Multi-Layer Data Integration Technique for Combining Heterogeneous Crime Data. Information Processing and Management, Vol. 59, 2022, No. 3, Art. No. 102879, doi: 10.1016/j.ipm.2022.102879.

[21] Ilhan, F.—Tekin, S. F.—Aksoy, B.: Spatio-Temporal Crime Prediction with Temporally Hierarchical Convolutional Neural Networks. 2020 28th Signal Processing and Communications Applications Conference (SIU), 2020, pp. 1–4, doi: 10.1109/SIU49456.2020.9302169.

[22] Wang, Q.—Jin, G.—Zhao, X.—Feng, Y.—Huang, J.: CSAN: A Neural Network Benchmark Model for Crime Forecasting in Spatio-Temporal Scale. Knowledge-Based Systems, Vol. 189, 2020, Art. No. 105120, doi: 10.1016/j.knosys.2019.105120.

[23] Chun, S. A.—Avinash Paturu, V.—Yuan, S.—Pathak, R.—Atluri, V.—Adam, N. R.: Crime Prediction Model Using Deep Neural Networks. Proceedings of the 20th Annual International Conference on Digital Government Research (dg.o 2019), ACM, 2019, pp. 512–514, doi: 10.1145/3325112.3328221.

[24] Krizhevsky, A.—Sutskever, I.—Hinton, G. E.: ImageNet Classification with Deep Convolutional Neural Networks. Communications of the ACM, Vol. 60, 2017, No. 6, pp. 84–90, doi: 10.1145/3065386.

[25] Gao, N.—Xue, H.—Shao, W.—Zhao, S.—Qin, K. K.—Prabowo, A.—Rahaman, M. S.—Salim, F. D.: Generative Adversarial Networks for Spatio-Temporal Data: A Survey. ACM Transactions on Intelligent Systems and Technology (TIST), Vol. 13, 2022, No. 2, Art. No. 22, doi: 10.1145/3474838.

[26] Prabowo, A.—Koniusz, P.—Shao, W.—Salim, F. D.: COLTRANE: ConvolutiOnaL TRAjectory NEtwork for Deep Map Inference. Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '19), 2019, pp. 21–30, doi: 10.1145/3360322.3360853.

[27] Alzubaidi, L.—Zhang, J.—Humaidi, A. J.—Al-Dujaili, A.—Duan, Y.—Al-Shamma, O.—Santamaría, J.—Fadhel, M. A.—Al-Amidie, M.—Farhan, L.: Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions. Journal of Big Data, Vol. 8, 2021, Art. No. 53, doi: 10.1186/s40537-021-00444-8.

[28] Borovykh, A.—Bohte, S.—Oosterlee, C. W.: Conditional Time Se-

ries Forecasting with Convolutional Neural Networks. CoRR, 2017, doi: 10.48550/arXiv.1703.04691.

[29] Zuo, R.—Xiong, Y.—Wang, J.—Carranza, E. J. M.: Deep Learning and Its Application in Geochemical Mapping. Earth-Science Reviews, Vol. 192, 2019, pp. 1–14, doi: 10.1016/j.earscirev.2019.02.023.

[30] Hoseinzade, E.—Haratizadeh, S.: CNNpred: CNN-Based Stock Market Prediction Using a Diverse Set of Variables. Expert Systems with Applications, Vol. 129, 2019, pp. 273–285, doi: 10.1016/j.eswa.2019.03.029.

[31] Barzegar, R.—Aalami, M. T.—Adamowski, J.: Short-Term Water Quality Variable Prediction Using a Hybrid CNN-LSTM Deep Learning Model. Stochastic Environmental Research and Risk Assessment, Vol. 34, 2020, No. 2, pp. 415–433, doi: 10.1007/s00477-020-01776-2.

[32] Chen, T.—Guestrin, C.: XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16), 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[33] Jiang, Y.—Tong, G.—Yin, H.—Xiong, N.: A Pedestrian Detection Method Based on Genetic Algorithm for Optimize XGBoost Training Parameters. IEEE Access, Vol. 7, 2019, pp. 118310–118321, doi: 10.1109/ACCESS.2019.2936454.

[34] Chen, Z.—Jiang, F.—Cheng, Y.—Gu, X.—Liu, W.—Peng, J.: XGBoost Classifier for DDoS Attack Detection and Analysis in SDN-Based Cloud. 2018 IEEE International Conference on Big Data and Smart Computing (BigComp), 2018, pp. 251–256, doi: 10.1109/BigComp.2018.00044.

[35] Alexandropoulos, S. A. N.—Kotsiantis, S. B.—Vrahatis, M. N.: Data Preprocessing in Predictive Data Mining. The Knowledge Engineering Review, Vol. 34, 2019, Art. No. e1, doi: 10.1017/S026988891800036X.

[36] Steffensmeier, D.—Harris, C. T.—Painter-Davis, N.: Gender and Arrests for Larceny, Fraud, Forgery, and Embezzlement: Conventional or Occupational Property Crime Offenders? Journal of Criminal Justice, Vol. 43, 2015, No. 3, pp. 205–217, doi: 10.1016/j.jcrimjus.2015.03.004.

[37] Sun, J.—Yue, M.—Lin, Z.—Yang, X.—Nocera, L.—Kahn, G.—Shahabi, C.: CrimeForecaster: Crime Prediction by Exploiting the Geographical Neighborhoods' Spatiotemporal Dependencies. In: Dong, Y., Ifrim, G., Mladenić, D., Saunders, C., Van Hoecke, S. (Eds.): Machine Learning and Knowledge Discovery in Databases. Applied Data Science and Demo Track (ECML PKDD 2020). Springer, Cham, Lecture Notes in Computer Science, Vol. 12461, 2021, pp. 52–67, doi: 10.1007/978-3-030-67670-4_4.

Ature Angbera is a current postgraduate student in the School of Computer Sciences, Universiti Sains, Malaysia. He also works with Joseph Sarwuan Tarkaa University Makurdi, Benue State in the Department of Computer Science. His main research fields are deep learning, machine learning, spatiotemporal analysis, stream processing, data mining and intelligent techniques.

Huah Yong Chan is Associate Professor in the School of Computer Sciences, Universiti Sains, Malaysia. His research fields are grid computing, cloud computing, resource allocation, and data analytics.

Computing and Informatics, Vol. 42, 2023, 591–614, doi: 10.31577/cai_2023_3_591

STEGANOGRAPHY APPROACH TO IMAGE AUTHENTICATION USING PULSE COUPLED NEURAL NETWORK

Radoslav Forgáč, Miloš Očkay

Institute of Informatics
Slovak Academy of Sciences
Dúbravská cesta 9
845 07 Bratislava, Slovakia
e-mail: {radoslav.forgac, milos.ockay}@savba.sk

Martin Javurek, Bianca Badidová

Department of Informatics
Armed Forces Academy of gen. M. R. Štefánik
Demänová 393
031 01 Liptovský Mikuláš, Slovakia
e-mail: {martin.javurek, bianca.badidova}@aos.sk

Abstract. This paper introduces a model for the authentication of large-scale images. The crucial element of the proposed model is the optimized Pulse Coupled Neural Network. This neural network generates position matrices based on which the embedding of authentication data into cover images is applied. Emphasis is placed on the minimalization of the stego image entropy change. Stego image entropy is consequently compared with the reference entropy of the cover image. The security of the suggested solution is granted by the neural network weights initialized with a steganographic key and by the encryption of accompanying steganographic data using the AES-256 algorithm. The integrity of the images is verified through the SHA-256 hash function. The integration of the accompanying and authentication data directly into the stego image and the authentication of the large images are the main contributions of the work.

Keywords: Image steganography, pulse coupled neural network, position matrix, image authentication

1 INTRODUCTION

Digital images are predominantly processed and transmitted in environments that allow for their modification. There is a growing need to develop procedures and methods that allow the integrity and authenticity of the image to be verified. Current examples mainly include industry, healthcare, military and many others. Image processing is greatly simplified by high-performance computing and specialized systems. Image modification is sophisticated and is used for a variety of purposes. An example is the misuse of artificial intelligence (AI) to generate fake images to commit fraud. Images generated by AI are so convincing that they can be difficult to distinguish from real images. The negative impacts of such manipulations might have far-reaching consequences on individuals or society. Copyright infringement or influencing public opinion can be mentioned, just to list a few examples. One of the possible solutions for authenticating images is to use steganographic methods. The term 'steganography' is derived from the Greek words 'steganos' (meaning hidden or concealed) and 'graphein' (meaning writing). Steganography proposes methods of data transmission using a carrier medium (cover media). The data transmission is implemented in such a way that it is not suspected that the information not directly related to the medium is being transmitted within the carrier medium. Suitable carrier media may be, for example, text, image, sound, video, etc. Analogously, any type of digital content, such as text, image, sound, video or binary code can be hidden using steganography. In order to extract the hidden information, the other party needs to know the steganographic method used to embed the message. It also concerns the accompanying data which are the parameters of the method used. Similar to cryptography, it is considered that the method for the embedding and extraction of the hidden message should be publicly available in order to verify the security of the method. The assurance of the untraceability of the hidden message in the cover medium should be a steganographic key, so-called stego key. Only knowledge of the stego key should lead to a successful extraction of the hidden content. Encryption is very often used to maximize the security of hidden content. In case of breaking the steganographic method, encryption is thus the final protection against the content being compromised.

This paper presents a steganographic method using position matrix generation by the Optimized Model of Pulse Coupled Neural Network (OM-PCNN). The position matrix can be considered as a prescription or template for embedding the hidden content into the carrier image. Due to the best setting of the position matrix, positions with high entropy are identified in the cover image. This is a prerequisite to eliminate the risk of detecting the positions of the hidden data in the cover image by the visual inspection. The security of the presented steganographic method is based on the secret stego key, the openness of the embedding algorithm and the extraction of the hidden message without the need to transmit the accompanying steganographic data by independent channel.

The rest of this paper is structured as follows. The second part describes related work on image authentication and steganography. In the third part, the theoretical background of the proposed model is described. The fourth part describes and evaluates the proposed authenticity verification model. The final part concludes the paper.

2 RELATED WORK

In recent years, there has been a rapid growth of research in the area of image data authentication [1, 2, 3]. There are two basic approaches for authentication: encryption-based approaches, i.e., cryptography and steganography-based approaches.

Another aspect of image authentication is the level of image modification. The first group consists of strict authentication methods which evaluate any image modification as inadmissible. Methods based on standard cryptography have been applied in this group [4]. Fragile watermarking-based methods have also been applied in this group, such as the method proposed based on the Frei-Chen edge mask [5], which provides excellent image quality with watermarking and clearly reveals tampered regions. Another example of a fragile watermarking-based method is a method combining discrete wavelet transform (DWT), singular value decomposition (SVD) and discrete cosine transform (DCT) [6]. The results showed that the method achieved high detection accuracy for various forms of modifications while maintaining high visual quality. The second group consists of selective authentication methods which tolerate selected operations on images, such as compression, various filtering algorithms, or the application of geometric transformations to images. Within the second group, semi-fragile watermarking methods have been applied, such as the Inner-Outer Block-Based method [7] which splits the image into an inner and an outer part. This division has the purpose of copyright protection in addition to authentication. The method has increased robustness to compression and common image operations such as gamma corrections, intensity adjustments and histogram equalization. Other methods for selective authentication include those using robust watermarking [8].

2.1 Cryptography-Based Image Authentication

Hashing functions are a key element of image authentication. A method to authenticate images using hashes with Multi-Attack Reference Generation and Adaptive Thresholding was proposed by [9]. The proposed method is based on clustering. A perceptual hashing algorithm was applied to the reference image to obtain the hash codes required for authentication. Adaptive thresholding was taken to account for variations in the hashing distance. The method showed a high performance but was rather time-consuming.

Tamper detection and localization in the images are still a subject of research. An approach was proposed by [10] in which the authors used hashes in combination

with fragile watermarks to authenticate and localize the tamper. In the presented approach, the original cover image is divided into non-overlapping blocks on which a DCT is performed. This operation extracts the coefficients to which the SHA-256 hash function is subsequently applied. After the hash is obtained, the original cover image is split into blocks by the Arnold transform using a key. A 16-bit hash is inserted into each block to obtain the watermarked image. The embedded hash function helps with tamper detection and image localization. The proposed approach was compared with several existing approaches, resulting in an improvement in peak signal-to-noise ratio (PSNR). Another study [11] also addressed image data authentication using hash functions, but the authors chose a neural network-based approach. A convolutional stacked denoising autoencoder was used for both authentication and tamper localization. The proposed autoencoder maps high-dimensional input data to hash codes. Then, the tamper localization is performed by comparing the decoder output of the tampered image with the hash of the real image. The authors used the F1-score metric and receiver operating characteristic (ROC) for evaluation. Results show better performance compared to other existing approaches.

Hashing as a stand-alone authentication tool is insufficient. Its primary function is to verify data integrity, i.e., that the data has not been modified in any way. Digital signatures [12, 13, 14] and Keyed-Hash Message Authentication Code [15, 16] are most commonly used to authenticate images.

Digital signature-based authentication has been addressed by [17] who proposed a novel image secret sharing (ISS) scheme. They introduced a two-way shadow image authentication method based on public key. The shadow image can be authenticated with the distributor's secret key in addition to the participants' private key. The proposed ISS scheme can decode secret images losslessly with bidirectional shadow image authentication without pixel expansion. The study in [18] proposed an asymmetric two-level phase generation image encryption scheme that uses a nonlinear decryption key generation process. The nonlinear encryption process provides a high level of resistance to the existing attacks. The use of a digital signature verifies the identity of the sender and no information is revealed without the use of the correct keys. An image encryption algorithm that hides a secret image and a digital signature that provides authenticity and confidentiality was proposed in [19]. The solution uses the Least Significant Bit (LSB) method to embed the digital signature and the Lifting Wavelet Transform (LWT) method to generate a meaningful encrypted image. Experimental results show that the proposed scheme has high key sensitivity. Based on the histogram analysis, it is found that the original carrier image and the final visual image are very similar.

Hash-based message authentication code (HMAC) was addressed in [20], in which the authors focused on the authentication of images from the healthcare domain. They proposed an optical algorithm that ensures the efficiency and security of medical image transmission. The proposed algorithm accomplished authentication and integrity by computing and verifying HMAC values. At the same time, confidentiality of medical images was achieved by using Rubik's cube encryption. The

effectiveness of the proposed algorithm has been thoroughly evaluated using various visual, qualitative, statistical and complex metrics. The security was evaluated by examining the key sensitivity and the robustness of the algorithm to different types of noise and attacks. Authentication using HMAC was also addressed by [21]. Their algorithm uses DCT combined with LSB. To verify the origin of the message, the HMAC of the transformed image is also embedded in the cover image. The proposed algorithm can identify data changes in the transmission channel but does not deal with the reconstruction of clipped or noisy images.

2.2 Authentication of Images Based on Data Hiding

Data hiding image authentication methods are based on digital steganography [22, 23, 24] or digital watermarking [25, 26, 27]. Currently, research in both sub-areas is also exploring the use of neural networks [28, 29, 30].

Digital watermarks are an ideal tool for verifying copyrighted images. A study in [31] proposes a blind dual watermarking scheme where the embedding of an invisible, robust watermark serves to protect the copyright and the embedding of a fragile watermark authenticates the image. The method in [32] focused on the use of blockchain to address the problem of trusted third parties protecting image copyrights. They tried to address the incompatibility between traditional digital watermarking technology and blockchain. They proposed a framework combining the zero-watermarking algorithm, the distributed storage system IPFS and the Ethereum blockchain. The proposed scheme has good robustness to noise filtering and moderate rotations.

Watermarks can be classified into several classes in terms of the monitored parameters. In terms of visual detection, we divide watermarks into visible [33, 34] and invisible [35, 36] watermarks. In terms of robustness to image transformations, we distinguish fragile watermarks, semi-fragile watermarks and robust watermarks. Fragile digital watermarks, like digital fingerprints, are very sensitive to virtually any transformations in the image, which is the main intention. A study in [37] used a dual fragile watermarking scheme to verify the integrity and localization of the tampered area. The results showed that the proposed scheme enhances the security of fragile watermarking and is robust to selected attacks. Semi-fragile watermarks are resilient to benign image operations but cannot handle significant operations. They are commonly used to detect significant image operations. A study in [38] used semi-fragile watermarks for authentication and tamper detection in the form of JPEG2000 compression. The proposed watermark generation process guides the system to verify the integrity of the image without the need for any other file except the watermarked image. Experimental results show that the proposed approach not only has extremely high tamper detection accuracy, but also has relatively high robustness to JPEG2000 compression. The last group is robust watermarking, which can withstand even significant image operations. Robust watermarks, in most cases, use frequency domain images for embedding. A study in [39] proposed an improved robust watermarking algorithm using discrete Fourier

transform (DFT) via spread spectrum that optimizes the number of bands and frequency coefficients, as well as the watermark strength factor using particle swarm optimization in conjunction with visual information fidelity and bit correct rate criteria. Experimental results show increased robustness to conventional signal processing and geometric distortions while maintaining the high visual quality of color images.

Steganography protects the hidden data in the cover image from detection [40, 41, 42]. Image steganography can be applied either directly in the spatial domain of images or in the frequency domain. Applications of steganography in the spatial domain have been addressed by [43]. The authors proposed a scheme using genetic algorithms to find optimal solutions. They used the LSB method for data embedding. To find the appropriate bits to hide the data, they used new concepts of shifting in vertical and horizontal directions, pixel scanning direction, secret image transposition, flipping of secret bits and using the XOR operation. The proposed scheme achieves high embedding capacity and reaches the desired imperceptibility of the stego image. Another study that addressed steganography in the spatial domain is [44], which proposed a multiple embedding scheme based on genetic algorithms for reversible data hiding based on histogram shifting. Compared with the previous approaches, experimental results show that the proposed scheme is superior in terms of embedding capacity and stego image quality. A study in [45] proposed a scheme using the integer wavelet transform (IWT) with improved embedding capacity. They used the coefficient value differencing (CVD) technique. The results showed that the eight-way CVD technique improves coefficient utilization, which helps to improve embedding capacity. Only high-frequency coefficients are used for embedding secret data because the distortion of high-frequency coefficients is less perceptible to the human eye than that of low-frequency coefficients. To enhance the security of the system, the secret data is embedded in horizontal and vertical subbands in a non-sequential manner. The proposed technique successfully resists both statistical and steganalysis attacks.

Steganography in the frequency domain was the subject of a study by [46]. The author used a secure medical data transmission mechanism based on a bit mask oriented genetic algorithm (BMOGA). The encrypted data is embedded in the medical images through 1-level and 2-level DWT. To extract the secret message from the encrypted one, the inverse process of BMOGA is implemented. The results show that the proposed algorithm is capable of secure data transmission. A study in [47] addressed the data hiding technique with enhanced embedding capacity using a combination of optimal pixel selection and LSB quantized DCT coefficients. It works with image partitioning into non-overlapping blocks of 8×8 pixels. The proposed scheme achieves high image quality because it selects the optimal pixels of a block. The performance of the proposed technique is evaluated using a standard dataset and compared with other state-of-the-art techniques. The proposed algorithm shows that the embedding capacity, stego image quality and processing time are better than other existing techniques.

3 THEORETICAL BACKGROUND OF PROPOSED MODEL

Our research in PCNNs began with the design of an optimized OM-PCNN model for reducing the dimension of the classification space. The OM-PCNN architecture is based on the original PCNN architecture with a modified feeding input [48, 49]. The optimization achieved a reduction in the number of parameters, furthermore, a mechanism for setting the initialization values of key parameters as well as an algorithm for generating features with a minimized number of iterations of the neural network were proposed [50, 51, 52, 53]. The structure of the original PCNN model and the OM-PCNN model itself are practically the same. It is a single-layer neural network. If we consider the input image as a 2D matrix, then the neural network matrix has the same structure as the image matrix. Each neuron has two defined inputs: a feeding input and a linking input (Figure 1).

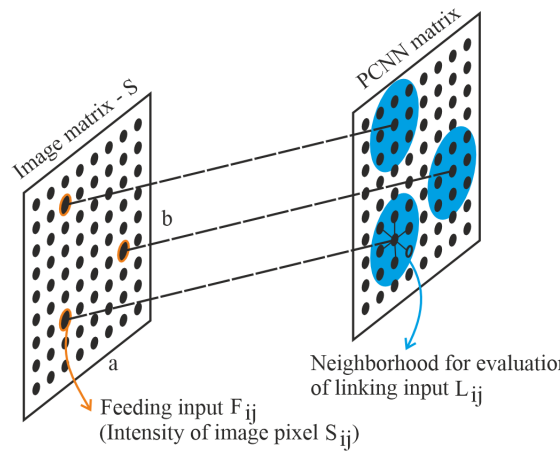


Figure 1. Structure of OM-PCNN neural network

The feeding input F_{ij} of each neuron is represented by one image pixel with intensity S_{ij} that corresponds positionally to the neuron in the neural network matrix. Linking input L_{ij} of each neuron depends on the number of active neurons in the linking neighborhood. Central neuron of each linking neighborhood will be reffered to as centroid. The size of the linking neighborhood, i.e., the OM-PCNN kernel matrix, depends on the linking radius r_o and the type of the neuron's linking neighborhood, which can be circular or square, depending on the used metric. Among the available metrics, the Chebyshev metric (C_D), Euclidean metric (E_D)

or Manhattan metric (M_D) are commonly applied:

$$C_D(x_i, x_j) = \max_k(|x_{ik} - x_{jk}|), \tag{1}$$

$$E_D(x_i, x_j) = \sqrt{\sum_{k=1}^{d}(x_{ik} - x_{jk})^2}, \tag{2}$$

$$M_D(x_i, x_j) = \sum_{k=1}^{d}|x_{ik} - x_{jk}|, \tag{3}$$

where d is the dimension of the feature space, x_i represents the centroid, and x_j is the neighborhood neuron.

The mathematical model of the OM-PCNN neuron (Figure 2) can be divided into three parts. The first part consists of the feeding and the linking input. The second part of the neuron contains the linking unit, in which the feeding and linking inputs are combined. The third part of the neuron is represented by a pulse generator and a threshold generator.

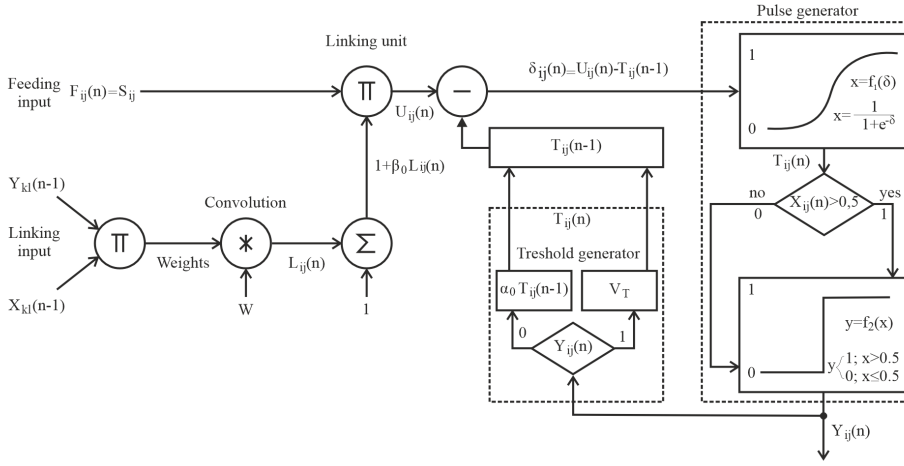


Figure 2. Mathematical model of the OM-PCNN neuron

The feeding input F_{ij} of centroid (i, j) in iteration n is represented by the following equation:

$$F_{ij}(n) = S_{ij}. \tag{4}$$

The linking input $L_{ij}(n)$ is given by the convolution of the weight matrix W and the product of the output matrices $X(n-1) \cdot Y(n-1)$ from the previous iteration $n-1$. The convolution is computed only for neurons that belong to the linking

neighborhood of the centroid (i, j):

$$L_{ij}(n) = [W * (X(n-1) \cdot Y(n-1))]_{ij}, \tag{5}$$

where the symbol "$*$" is the convolution operator and the symbol "$\cdot$" is the multiplication operator of the output matrices $X(n-1)$ and $Y(n-1)$. The elements of the matrix $X(n-1)$ represent the output activation quantities of the corresponding neurons based on the sigmoidal activation function (8) from the previous iteration $n-1$. The elements of the matrix $Y(n-1)$ represent the output activation quantities based on the step activation function (9) from the previous iteration $n-1$. Each element of the kernel matrix W represents the connection weight between the centroid and the neighborhood neuron. The elements of the kernel matrix W, i.e., the values of the weight coefficients, depend on the link radius r_o and the implemented kernel of the neural network. The most commonly used kernels for PCNNs are the kernel based on the Gaussian distribution or the $1/r$, $1/r^2$ kernels. In the case of using OM-PCNN for steganography, the kernel is generated using a stego key.

In the linking part of the neuron, occurs the modulation of the feeding input S_{ij} with the linking element $(1 + \beta_o \cdot L_{ij}(n))$. The modulation result is the input potential of the neuron $U_{ij}(n)$, which can be characterized by the formula:

$$U_{ij}(n) = S_{ij} \cdot [1 + \beta_0 \cdot L_{ij}(n)], \tag{6}$$

where β_o is the linking coefficient, which determines the degree of modulation of the feeding and linking inputs.

The level of the neuron's action potential $\delta_{ij}(n)$ has a profound effect on the neuron's pulsation. Pulsation is the output effect of each neuron in the OM-PCNN representing a series of active and inactive states of the neuron in a time sequence. The action potential $\delta_{ij}(n)$ is given by the difference of the neuron's current input potential $U_{ij}(n)$ and the threshold potential $T_{ij}(n-1)$ from the previous iteration of the OM-PCNN. $U_{ij}(n)$ has activating effect, while $T_{ij}(n-1)$ has inhibitory effect on neuron activity:

$$\delta_{ij}(n) = U_{ij}(n) - T_{ij}(n-1). \tag{7}$$

In the third part of the OM-PCNN neuron, it is decided whether the neuron will be activated or not. The third part of the neuron is made of a pulse generator and a threshold generator. In the pulse generator, a sigmoidal activation function evaluates the first neuron's output $X_{ij}(n)$, which determines the degree of activation of the neuron:

$$X_{ij}(n) = \frac{1}{1 + e^{-\delta_{ij}(n)}}. \tag{8}$$

The values of $X_{ij}(n)$ are in the interval $\langle 0, 1 \rangle$. The second neuron's output $Y_{ij}(n)$ depends on $X_{ij}(n)$ and determines whether the neuron in iteration n is active. The

output $Y_{ij}(n)$ is based on the step activation function:

$$Y_{ij}(n) = \begin{cases} 1, & \text{if } X_{ij}(n) > 0.5, \\ 0, & \text{else.} \end{cases} \tag{9}$$

The step activation function normalizes the output of each neuron to the binary values 0 (neuron activation is suppressed) and 1 (neuron is activated). This is the basic principle of generating binary images using OM-PCNN in each iteration step n. Based on the value of $Y_{ij}(n)$, the threshold potential of the neuron $T_{ij}(n)$ is then calculated:

$$T_{ij}(n) = \begin{cases} V_T, & \text{if } Y_{ij}(n) = 1, \\ \alpha_o \cdot T_{ij}(n-1), & \text{if } Y_{ij}(n) = 0, \end{cases} \tag{10}$$

where the parameter α_o is the threshold decay coefficient and the parameter V_T is the threshold potential coefficient. Formulas (4), (5), (6), (7), (8), (9) and (10) represent one iteration of the neuron. The number of iterations N, in most cases, is given by a qualified guess. In the case of OM-PCNN for steganographic purposes, it is in the interval $\langle 1, 5 \rangle$.

It has been shown that OM-PCNN has a potential in the field of image steganography, mainly due to its robustness to noise [54]. OM-PCNN generates a series of temporary binary images that represent the current state of the neurons in a given iteration (Figure 3). These binary images are the candidate position matrices for embedding.

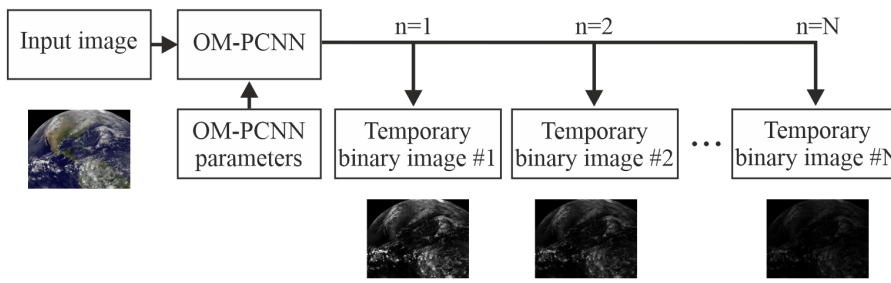


Figure 3. A series of binary images generated using OM-PCNN

The position matrices serve as templates for embedding messages into the cover images. The individual bits of the hidden message will be inserted into the cover image according to selected position matrix. Each binary image generated by the OM-PCNN within the n^{th} iteration can be considered a position matrix for the placement of the hidden message if it satisfies two basic criteria. The first criterion is the complete matching of the binary matrices of the cover image and the stego image. The second criterion is the capacity of the position matrix, which must be at least equal to the size of the hidden message. The selection of binary image matrices

within the first iteration, i.e., $n = 1$, is not recommended due to the initialization of the OM-PCNN. The embedding itself is performed using the LSB method, which is one of the most commonly used methods in the spatial domain of images. The principle of embedding the hidden message into the image itself is based on the LSB method (Figure 4). The bits of the hidden message are inserted only in those image points of the cover image that correspond to the unit elements of the position matrix:

$$y_i = \begin{cases} x_i + 1, \\ x_i - 1, \\ x_i, \end{cases} \tag{11}$$

where x_i is the original pixel value and y_i is the modified pixel value after LSB substitution. OM-PCNN-based steganography has two major advantages, namely the generation of a position matrix for message embedding and the invariance of OM-PCNN to noise.

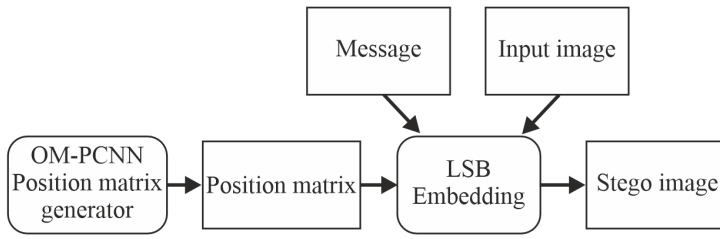


Figure 4. The embedding principle using OM-PCNN and LSB

4 AUTHENTICITY VERIFICATION MODEL

Our image authentication approach is based on a combination of cryptography (hashing and symmetric encryption), steganography and OM-PCNN neural network. The main goal was to design an offline solution where there is no need to store the accompanying steganographic data in secure repositories. The presented method is based on a unique stego key. The description of the detailed protocol for generating the different elements (stego key, random number, kernel, etc.) and the procedures in the authentication process is beyond the scope of this paper. Based on the description of the proposed method, one can proceed with the actual protocol specification, e.g., selection of the hashing method, random number generator, encryption algorithm, generation of starting positions, etc.

The testing set consisted of 500 gray satellite images with 8 bit depth without compression. The resolution $(x \times y)$ of those images was $1\,000 \times 1\,000$, $2\,000 \times 2\,000$ and $3\,000 \times 3\,000$. All experiments were realized within the inner matrix (IM) with the resolution $(a \times b)$ of 500×500 pixels (Figure 5).

Our authenticity verification concept consists of a proposed model baseline, cover image protection and the final stego image authenticity verification.

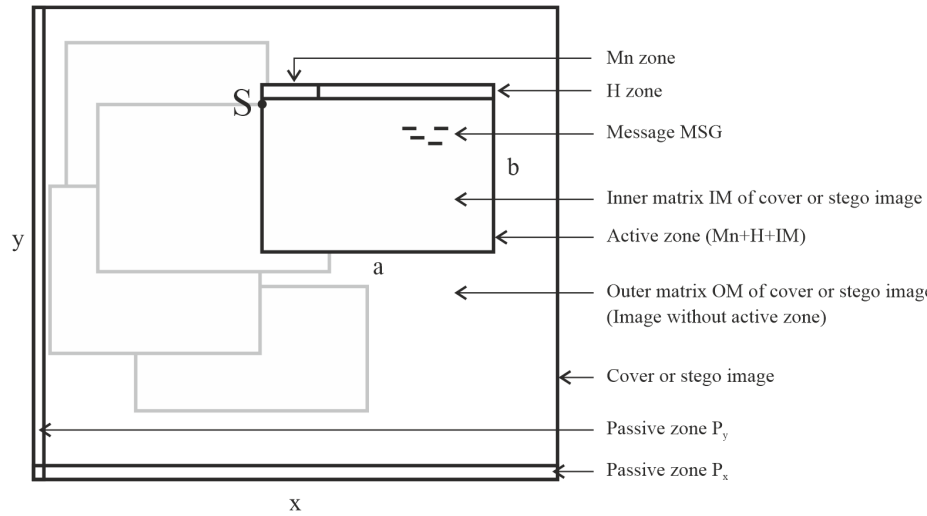


Figure 5. Cover or stego image with embedding zones

4.1 Baseline of the Proposed Model

For descriptive purposes, we have named the entity to insert the message into the cover image "Publisher" and the entity to extract the message from the cover image "Recipient". The success of the proposed steganographic method is dependent on several factors. OM-PCNN is the key factor of the proposed model. The optimal parameter setting allows to generate identical position matrices from both the cover image and the stego image. This means that the same position matrix is used for message embedding into the cover image at the Publisher side and for extracting the message from the stego image at the Recipient side. Message embedding and extraction are two independent processes. In [55] the influence of OM-PCNN parameters on the generation of position matrices is described. Namely, linking coefficient – β_o, linking radius – r_o, Type of OM-PCNN kernel – K, threshold decay coefficient – α_o, threshold potential coefficient – V_T, initialization value of threshold potential – $T(0)$, the number of iterations per cycle – N and type of activation function are explained. In the interval of 2 to 5 iterations, for which the OM-PCNN is optimized, the key influence of the parameter pair α_o and $T(0)$ has been demonstrated. In [56], two approaches for evaluating the quality of stego images based on entropy were compared. The first approach is based on the OM-PCNN position matrix. The second approach was based on generating random positions for embedding. Experiments

showed that embedding using a position matrix is a more efficient method compared to random embedding. OM-PCNN allows to locate regions with higher entropy in the images, thus minimizing the probability of embedding detection.

A proposed method to authenticate large images is based on the work of [57]. Experiments with cover images with a resolution of 500×500 pixels evoked the idea of applying "window" steganography. The baseline of this method is a window selection from the original large cover image, in which operations are performed to ensure the authenticity of the whole image. Compared to the original model, all the accompanying data required for message extraction at the Receiver side is part of the stego image. The principle of output parameter minimization applies, i.e., using only the necessary parameters to extract the hidden messages from the stego images.

In the preparatory phase, the necessary operations for the authentication process need to be carried out:

1. Generating stego key. The stego key is generated from the password using the SHA-256 hash function. The stego key is distributed between the Publisher and the Recipient only.
2. Unique number (UN) generation. This number is unique for each cover image. Our protocol uses a string length of 256 bits.
3. Generation of the start positions of the OM-PCNN key-parameter interval searches – α_o, $T(0)$. These positions are computed by combining the stego key and the pixel values of the passive zone P_x and P_y.
4. OM-PCNN kernel generation. The kernel represents the weight matrix of the neural network, which is computed by combining the stego key and the pixel values of the passive zone P_x and P_y.
5. Symmetric encryption key generation is optional to make the contents of sensitive data inaccessible in case of breaking the steganographic method. Our protocol uses AES-256 with a key equal to stego key.

4.2 Cover Image Protection

In this work, the term "cover image protection" refers to the creation of a stego image with implemented protection mechanisms. The pseudo-algorithm on the Publisher side can be described as follows:

1. The Publisher generates a 256-bit hash code from the outer cover image matrix. The hash is concatenated with the UN to produce a message of length 512 bits (MSG). The reason for combining the hash and the UN is to reduce the dependency of the MSG solely on the image data.
2. The Publisher generates the position of the inner image matrix (IM) based on the stego key and the passive zone P_x for x-axis, P_y for y-axis of the cover image (P_x + stego key, P_y + stego key). This means that the IM position will

be different for each image. The IM must not interfere with the passive zone. This is due to the possible overwriting of some bits in the passive zone caused by MSG embedding.

3. For the adaptation of the key parameters of the OM-PCNN, we seek a combination of parameters in such a way that the neural network generates the same position matrix for both the cover image and the stego image. The number of cycles M completed to find a suitable combination and the iteration number n within the final cycle are part of the accompanying data, which we insert into the Mn zone after AES-256 encryption with stego key. Using the values of M and n, we can compute the key parameters α_o, $T(0)$ to generate the position matrix. During adaptation, at each cycle and iteration, the candidate position matrix of the cover image is compared with the corresponding candidate stego image. If the position matrix candidates match, the position matrix and also the key parameters α_o, $T(0)$ have been found. The starting position of the MSG embedding is given by the centroid of the position matrix.
4. The Publisher inserts the MSG using the position matrix from point 3 into the IM. The result represents the inner stego image.
5. The Publisher adds a UN to the end of the inner stego image matrix and generates a 256-bit hash. The hash is inserted into the H zone. The generation of the stego image of the original cover image is complete.
6. The original cover image is deleted or stored in a protected location by the Publisher so that the cover image and stego image matrices cannot be compared. The stego image is considered to be the original.

4.3 Stego Image Authenticity Verification

The pseudo-algorithm corresponds to the process of extracting the MSG from the stego image on the Recipient side:

1. The Receiver calculates the IM position in the stego image. It decrypts the values of M and n from the Mn zone using AES-256 with stego key. In case of a decryption error, the authenticity of the stego image can be violated. After successful decryption, the starting position of the parameters α_o, $T(0)$ is determined using the stego key and the passive zone of the stego image. The values of the parameters α_o, $T(0)$ are calculated using M.
2. The Receiver generates the OM-PCNN kernel using stego key and the pixel values of the passive zone P_x and P_y. The IM is fed to the input of the OM-PCNN and the position matrix for the n^{th} iteration is generated. The centroid position of the position matrix is computed and the MSG is extracted.
3. The Receiver generates and compares the OM hash with the hash from the MSG. If the hashes match, the outer image matrix is authentic.

4. The Receiver adds a UN from MSG to the end of the IM and generates a 256-bit hash. If the generated hash matches the hash in the H zone, the stego image is authentic.

4.4 Evaluation of the Model

It is well known that PCNNs, due to their iterative nature are more time-consuming to process images. The random starting position of scanning the intervals of key parameters of OM-PCNNs can speed up the parameter adaptation significantly, but on the contrary, it can also slow it down. Parameter adaptation with step 0.01 requires about 1 500 cycles for a complete search of the redefined ranges of the two key parameters, which in the presented model represents up to 7 500 iterations for a single image in the worst case. The situation worsened with the increasing size of the images. For example, the processing of a 500×500 resolution image is about 3.6 times slower than a 200×200 resolution image. The idea of implementing steganography in a predefined cover image cutout unifies the computational complexity of generating position matrices for any large image solely based on the size of the image matrix of that cutout. The question is whether the limited size of the image matrix for embedding will be capaciously sufficient for authentication purposes. The experimental results clearly demonstrated that for a test set of images with a resolution of 500×500, a minimum embedding capacity of 723 bytes and a maximum capacity of up to 31 237 bytes were achieved (Figure 6). This is the capacity at which OM-PCNN can generate identical position matrices for both the cover image and the stego image, which is a prerequisite for the successful deployment of the presented method. This means that even images with minimal embedding capacity offer about three times more space than required by the authenticity itself, including the accompanying data.

The quality of the steganographic method can be determined by the change in the entropy of the image after the message is embedded. The entropy calculation is given by the formula

$$H(x) = -\sum_{i \in 0}^{255} p_i \log_2 p_i, \tag{12}$$

where i is the pixel intensity value and p_i is the probability of these pixel values occurring in the image. The maximum achievable entropy for images with a bit depth of 8 bits per pixel is 8. The entropy change after OM-PCNN based embedding and random MSG distribution in the image has been tested. The reference value is the entropy of the cover image. The images were divided into five groups according to the maximum embedding capacity (Figure 7). The results show that the lowest entropy change was achieved within each group using OM-PCNN based embedding (Table 1).

Despite the positive results achieved above, the presented model also has weaknesses. The problem is the embedding of the accompanying data (see Section 4.2, No. 3) in the Mn zone and the hashes (see Section 4.2, No. 5) in the H zone. These

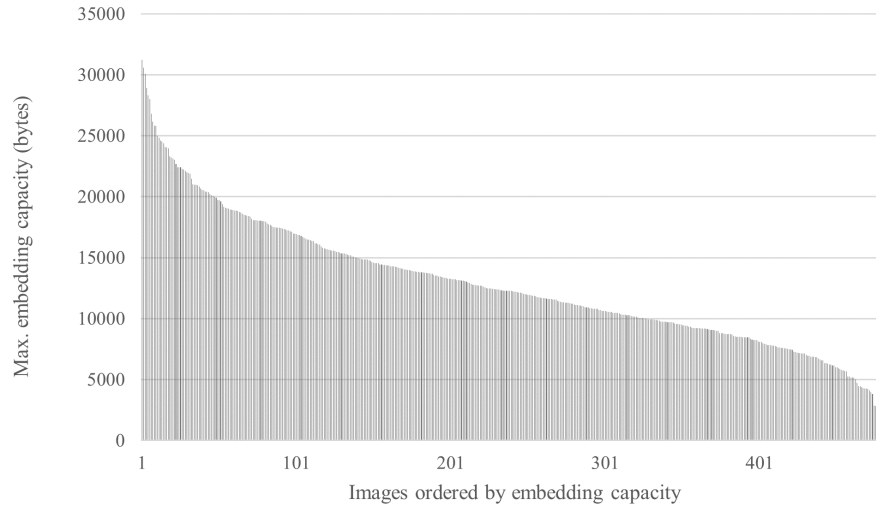


Figure 6. Overview of the maximum embedding capacity achieved for a group of 500 images with a resolution of 500×500

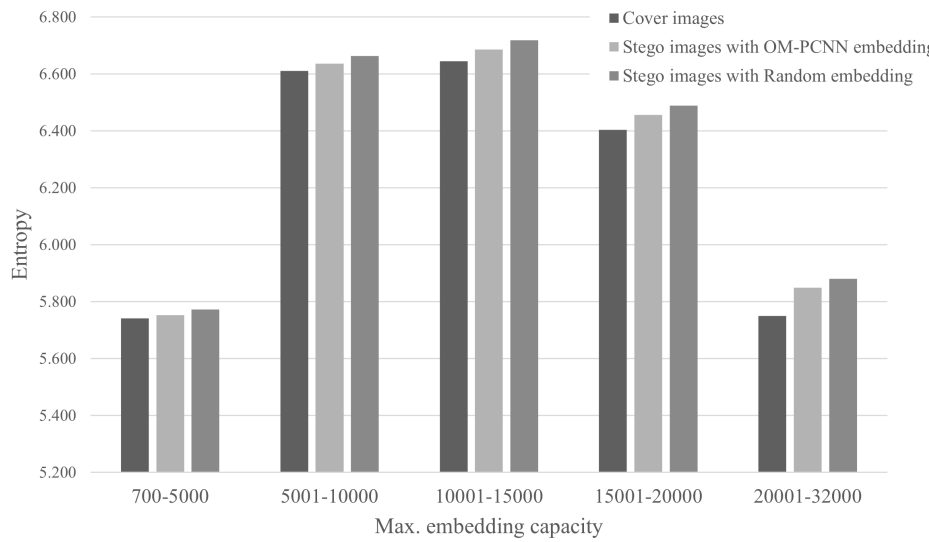


Figure 7. Overview of entropy values for five groups of images according to the maximum achieved embedding capacity

Maximum Embedding Capacity Range (Bytes)	Entropy of Cover Images	Entropy Change (%)	
		OM-PCNN Approach	Random Embedding
700–5 000	5.740	0.20	0.55
5 001–10 000	6.610	0.38	0.80
10 001–15 000	6.645	0.60	1.11
15 001–20 000	6.403	0.82	1.33
20 001–32 000	5.749	1.73	2.28

Table 1. Change of entropy by maximum embedding

are about 450 bits that are stored sequentially over the IM matrix, which may increase the probability of detectability of the embedded data. Since the security measure of the steganographic method is of primary importance in authentication, i.e., the unbreakability of the data embedding and extraction algorithms, the risk of steganography detection can be acceptable. This means that despite the detection of the embedding positions and the subsequent modification attempt in the Mn zone, the encrypted accompanying data will be degraded, which is evaluated by the system as an authenticity violation. Similarly, any modification in the H zone will trigger a hash mismatch on the Receiver side. Moreover, the position of the IM image matrix is different for each image as it depends on the stego key, the passive zone of the cover and the stego image, respectively. There may be other hidden risks that could be revealed by detailed steganalysis.

5 CONCLUSIONS

The main goal of any steganographic method is to minimize the probability of detecting or suspecting the existence of a hidden message in the stego image. In the case of authentication by steganography, however, this is not always a requirement. Our proposed model provides an efficient way to ensure the integrity and authenticity of high-resolution gray images relatively quickly. To minimize the processing time of large images, steganography has been proposed only in the inner image matrix, which is a subset of the original cover image. The presented OM-PCNN based method embeds the required data in the regions of high entropy. The entropy change after message embedding is lower compared to random embedding. The above attributes reduce the detection probability of embedded authentication data and accompanying data. Another advantage of the presented solution is the extension of the embedded data types. In addition to authentication data, other data related to a particular image can be embedded. For example, this can be personal data, access permissions, identifiers, passwords, or even data used to annotate the images, which can be used to search and sort the images according to different criteria. The advantage of the steganographic approach is that the data is an integral part of the subject image and is only accessible based on knowledge of the stego key.

The proposed steganography model belongs to the category of strict image authentication methods. It can effectively determine whether an image has been altered, even if it is a single pixel modification. Although the model does not allow the change to be localized, this is not a necessary requirement in the case of authentication. The presented model is not suitable for the authentication of images for which geometric transformations or conversion to another image format must be performed. These operations are evaluated by the model as modifications that corrupt the integrity of the image data. The strictness of the method can also be an issue for non-substantial image modifications, such as bitwise modifications in non-validated transmission protocols or automatic modification of image resolution during transmission using some communication applications. Model security requires detailed steganalysis. In the case of avoiding steganalytic detection and hidden message extraction, the option to encrypt all embedded data is still available.

In conclusion, the presented model based on the steganographic method using the OM-PCNN neural network has wide implementation possibilities. In the near future, it is planned to be included in a system for anomaly detection in distributed systems as well as for annotation of image data in order to determine the prevailing visibility.

Acknowledgement

This work was supported by the Slovak Research and Development Agency under the Contract No. APVV-20-0571 (ICONTROL) and by the Slovak Scientific Grant Agency VEGA 2/0131/23.

REFERENCES

[1] Shaik, A. S.—Karsh, R. K.—Islam, M.—Laskar, R. H.: A Review of Hashing Based Image Authentication Techniques. Multimedia Tools and Applications, Vol. 81, 2022, No. 2, pp. 2489–2516, doi: 10.1007/s11042-021-11649-7.

[2] Raj, N. R. N.—Shreelekshmi, R.: A Survey on Fragile Watermarking Based Image Authentication Schemes. Multimedia Tools and Applications, Vol. 80, 2021, No. 13, pp. 19307–19333, doi: 10.1007/s11042-021-10664-y.

[3] Chennamma, H. R.—Madhushree, B.: A Comprehensive Survey on Image Authentication for Tamper Detection with Localization. Multimedia Tools and Applications, Vol. 82, 2022, No. 2, doi: 10.1007/s11042-022-13312-1.

[4] Du, L.—Ho, A. T. S.—Cong, R.: Perceptual Hashing for Image Authentication: A Survey. Signal Processing: Image Communication, Vol. 81, 2020, Art. No. 115713, doi: 10.1016/j.image.2019.115713.

[5] Renklier, A.—Öztürk, S.: A Novel Frei-Chen Based Fragile Watermarking Method for Authentication of an Image. Concurrency and Computation: Practice and Experience, Vol. 34, 2022, No. 22, Art. No. e6897, doi: 10.1002/cpe.6897.

[6] Nguyen, T. S.: Fragile Watermarking for Image Authentication Based on DWT-SVD-DCT Techniques. Multimedia Tools and Applications, Vol. 80, 2021, No. 16, pp. 25107–25119, doi: 10.1007/s11042-021-10879-z.

[7] Senol, A.—Elbasi, E.—Topcu, A. E.—Mostafa, N.: A Semi-Fragile, Inner-Outer Block-Based Watermarking Method Using Scrambling and Frequency Domain Algorithms. Electronics, Vol. 12, 2023, No. 4, Art. No. 1065, doi: 10.3390/electronics12041065.

[8] Kadian, P.—Arora, S. M.—Arora, N.: Robust Digital Watermarking Techniques for Copyright Protection of Digital Data: A Survey. Wireless Personal Communications, Vol. 118, 2021, No. 4, pp. 3225–3249, doi: 10.1007/s11277-021-08177-w.

[9] Du, L.—He, Z.—Wang, Y.—Wang, X.—Ho, A. T. S.: An Image Hashing Algorithm for Authentication with Multi-Attack Reference Generation and Adaptive Thresholding. Algorithms, Vol. 13, 2020, No. 9, Art. No. 227, doi: 10.3390/a13090227.

[10] Hussan, M.—Parah, S. A.—Jan, A.—Qureshi, G. J.: Hash-Based Image Watermarking Technique for Tamper Detection and Localization. Health and Technology, Vol. 12, 2022, No. 2, pp. 385–400, doi: 10.1007/s12553-021-00632-9.

[11] Shaik, A. S.—Karsh, R. K.—Islam, M.—Singh, S. P.: A Secure and Robust Autoencoder-Based Perceptual Image Hashing for Image Authentication. Wireless Communications and Mobile Computing, Vol. 2022, 2022, Art. No. 1645658, doi: 10.1155/2022/1645658.

[12] Gafsi, M.—Amdouni, R.—Hajjaji, M. A.—Malek, J.—Mtibaa, A.: Improved Chaos-RSA-Based Hybrid Cryptosystem for Image Encryption and Authentication. Concurrency and Computation: Practice and Experience, Vol. 34, 2022, No. 23, Art. No. e7187, doi: 10.1002/cpe.7187.

[13] Jasra, B.—Moon, A. H.: Color Image Encryption and Authentication Using Dynamic DNA Encoding and Hyper Chaotic System. Expert Systems with Applications, Vol. 206, 2022, Art. No. 117861, doi: 10.1016/j.eswa.2022.117861.

[14] Parida, P.—Pradhan, C.—Gao, X. Z.—Roy, D. S.—Barik, R. K.: Image Encryption and Authentication with Elliptic Curve Cryptography and Multidimensional Chaotic Maps. IEEE Access, Vol. 9, 2021, pp. 76191–76204, doi: 10.1109/ACCESS.2021.3072075.

[15] Wu, X.—Yang, C. N.—Yang, Y. Y.: Sharing and Hiding a Secret Image in Color Palette Images with Authentication. Multimedia Tools and Applications, Vol. 79, 2020, No. 35-36, pp. 25657–25677, doi: 10.1007/s11042-020-09253-2.

[16] Hlaing, A. T.—Thant, K. M.: Color Image Steganography Using Cryptography and Magic LSB Substitution Method (M-LSB-SM). 2018 Joint International Conference on Science, Technology and Innovation, IEEE, 2019.

[17] Yan, X.—Li, L.—Chen, J.—Sun, L.: Public Key Based Bidirectional Shadow Image Authentication Without Pixel Expansion in Image Secret Sharing. Frontiers of Information Technology and Electronic Engineering, Vol. 24, 2023, No. 1, pp. 88–103, doi: 10.1631/FITEE.2200118.

[18] Khurana, M.—Singh, H.: Two Level Phase Retrieval in Fractional Hartley Domain for Secure Image Encryption and Authentication Using Digital Signatures. Multimedia Tools and Applications, Vol. 79, 2020, No. 19, pp. 13967–13986, doi:

10.1007/s11042-020-08658-3.

[19] Huang, X.—Dong, Y.—Ye, G.—Yap, W. S.—Goi, B. M.: Visually Meaningful Image Encryption Algorithm Based on Digital Signature. Digital Communications and Networks, Vol. 9, 2023, No. 1, pp. 159–165, doi: 10.1016/j.dcan.2022.04.028.

[20] El-Shafai, W.—Almomani, I.—Ara, A.—Alkhayer, A.: An Optical-Based Encryption and Authentication Algorithm for Color and Grayscale Medical Images. Multimedia Tools and Applications, Vol. 82, 2023, No. 15, pp. 23735–23770, doi: 10.1007/s11042-022-14093-3.

[21] Sheidaee, A.—Farzinvash, L.: A Novel Image Steganography Method Based on DCT and LSB. 2017 9th International Conference on Information and Knowledge Technology (IKT), Tehran, Iran, 2017, pp. 116–123, doi: 10.1109/IKT.2017.8258628.

[22] Muhammad, K.—Ahmad, J.—Rho, S.—Baik, S. W.: Image Steganography for Authenticity of Visual Contents in Social Networks. Multimedia Tools and Applications, Vol. 76, 2017, No. 18, pp. 18985–19004, doi: 10.1007/s11042-017-4420-8.

[23] Alarood, A.—Ababneh, N.—Al-Khasawneh, M.—Rawashdeh, M.—Al-Omari, M.: IoTSteg: Ensuring Privacy and Authenticity in Internet of Things Networks Using Weighted Pixels Classification Based Image Steganography. Cluster Computing, Vol. 25, 2022, No. 3, pp. 1607–1618, doi: 10.1007/s10586-021-03383-4.

[24] Gutub, A.—Al-Ghamdi, M.: Hiding Shares by Multimedia Image Steganography for Optimized Counting-Based Secret Sharing. Multimedia Tools and Applications, Vol. 79, 2020, No. 11, pp. 7951–7985, doi: 10.1007/s11042-019-08427-x.

[25] Sharma, S.—Zou, J. J.—Fang, G.: A Single Watermark Based Scheme for Both Protection and Authentication of Identities. IET Image Processing, Vol. 16, 2022, No. 12, pp. 3113–3132, doi: 10.1049/ipr2.12542.

[26] Sanivarapu, P. V.—Rajesh, K. N. V. P. S.—Hosny, K. M.—Fouda, M. M.: Digital Watermarking System for Copyright Protection and Authentication of Images Using Cryptographic Techniques. Applied Sciences, Vol. 12, 2022, No. 17, Art. No. 8724, doi: 10.3390/app12178724.

[27] Wang, Y.—Li, Z.—Gong, D.—Lu, H.—Liu, F.: Image Fragile Watermarking Algorithm Based on Deneighbourhood Mapping. IET Image Processing, Vol. 16, 2022, No. 10, pp. 2652–2664, doi: 10.1049/ipr2.12515.

[28] Jarusek, R.—Volna, E.—Kotyrba, M.: Photomontage Detection Using Steganography Technique Based on a Neural Network. Neural Networks, Vol. 116, 2019, pp. 150–165, doi: 10.1016/j.neunet.2019.03.015.

[29] Zhang, S.—Su, S.—Li, L.—Lu, J.—Zhou, Q.—Chang, C. C.: CSST-Net: An Arbitrary Image Style Transfer Network of Coverless Steganography. The Visual Computer, Vol. 38, 2022, No. 6, pp. 2125–2137, doi: 10.1007/s00371-021-02272-6.

[30] Hussain, S.—Sheybani, N.—Neekhara, P.—Zhang, X.—Duarte, J.—Koushanfar, F.: FastStamp: Accelerating Neural Steganography and Digital Watermarking of Images on FPGAs. Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design (ICCAD '22), ACM, 2022, Art. No. 41, doi: 10.1145/3508352.3549357.

[31] Ahmadi, S. B. B.—Zhang, G.—Rabbani, M.—Boukela, L.—Jelodar, H.: An Intelligent and Blind Dual Color Image Watermarking for Authentication and

Copyright Protection. Applied Intelligence, Vol. 51, 2021, No. 3, pp. 1701–1732, doi: 10.1007/s10489-020-01903-0.

[32] Wang, B.—Jiawei, S.—Wang, W.—Zhao, P.: Image Copyright Protection Based on Blockchain and Zero-Watermark. IEEE Transactions on Network Science and Engineering, Vol. 9, 2022, No. 4, pp. 2188–2199, doi: 10.1109/TNSE.2022.3157867.

[33] Fragoso-Navarro, E.—Rangel-Espinoza, K.—Nakano-Miyatake, M.—Cedillo-Hernandez, M.—Perez-Meana, H.: Seam Carving Based Visible Watermarking Robust to Removal Attacks. Journal of King Saud University – Computer and Information Sciences, Vol. 34, 2022, No. 7, pp. 4499–4513, doi: 10.1016/j.jksuci.2021.03.010.

[34] Qi, W.—Liu, Y.—Guo, S.—Wang, X.—Guo, Z.: An Adaptive Visible Watermark Embedding Method Based on Region Selection. Security and Communication Networks, Vol. 2021, 2021, Art. No. 6693343, doi: 10.1155/2021/6693343.

[35] Liu, C.—Zhong, D.—Shao, H.: Data Protection in Palmprint Recognition via Dynamic Random Invisible Watermark Embedding. IEEE Transactions on Circuits and Systems for Video Technology, Vol. 32, 2022, No. 10, pp. 6927–6940, doi: 10.1109/TCSVT.2022.3174582.

[36] Liang, J.—Feng, Z.—Chen, R.—Liu, X.: BHI: Embedded Invisible Watermark as Adversarial Example Based on Basin-Hopping Improvement. Information Sciences, Vol. 640, 2023, Art. No. 119037, doi: 10.1016/j.ins.2023.119037.

[37] Gong, X.—Chen, L.—Yu, F.Zhao, X.—Wang, S.: A Secure Image Authentication Scheme Based on Dual Fragile Watermark. Multimedia Tools and Applications, Vol. 79, 2020, No. 25, pp. 18071–18088, doi: 10.1007/s11042-019-08594-x.

[38] Rhayma, H.—Makhloufi, A.—Hamam, H.—Hamida, A. B.: Semi-Fragile Watermarking Scheme Based on Perceptual Hash Function (PHF) for Image Tampering Detection. Multimedia Tools and Applications, Vol. 80, 2021, No. 17, pp. 26813–26832, doi: 10.1007/s11042-021-10886-0.

[39] Cedillo-Hernandez, M.—Cedillo-Hernandez, A.—Garcia-Ugalde, F. J.: Improving DFT-Based Image Watermarking Using Particle Swarm Optimization Algorithm. Mathematics, Vol. 9, 2021, No. 15, Art. No. 1795, doi: 10.3390/math9151795.

[40] Li, G.—Feng, B.—He, M.—Weng, J.—Lu, W.: High-Capacity Coverless Image Steganographic Scheme Based on Image Synthesis. Signal Processing: Image Communication, Vol. 111, 2023, Art. No. 116894, doi: 10.1016/j.image.2022.116894.

[41] Gao, K.—Chang, C. C.—Horng, J. H.—Echizen, I.: Steganographic Secret Sharing via AI-Generated Photorealistic Images. EURASIP Journal on Wireless Communications and Networking, Vol. 2022, 2022, Art. No. 119, doi: 10.1186/s13638-022-02190-8.

[42] Chowdhuri, P.—Pal, P.—Si, T.: A Novel Steganographic Technique for Medical Image Using SVM and IWT. Multimedia Tools and Applications, Vol. 82, 2023, No. 13, pp. 20497–20516, doi: 10.1007/s11042-022-14301-0.

[43] Wazirali, R.—Alasmary, W.—Mahmoud, M. M. E. A.—Alhindi, A.: An Optimized Steganography Hiding Capacity and Imperceptibly Using Genetic Al-

gorithms. IEEE Access, Vol. 7, 2019, pp. 133496–133508, doi: 10.1109/ACCESS.2019.2941440.

[44] Wang, J.—Ni, J.—Zhang, X.—Shi, Y. Q.: Rate and Distortion Optimization for Reversible Data Hiding Using Multiple Histogram Shifting. IEEE Transactions on Cybernetics, Vol. 47, 2017, No. 2, pp. 315–326, doi: 10.1109/TCYB.2015.2514110.

[45] Mandal, P. C.—Mukherjee, I.—Chatterji, B. N.: High Capacity Steganography Based on IWT Using Eight-Way CVD and n-LSB Ensuring Secure Communication. Optik, Volume 247, 2021, Art. No. 167804, doi: 10.1016/j.ijleo.2021.167804.

[46] Pandey, H. M.: Secure Medical Data Transmission Using a Fusion of Bit Mask Oriented Genetic Algorithm, Encryption and Steganography. Future Generation Computer Systems, Vol. 111, 2020, pp. 213–225, doi: 10.1016/j.future.2020.04.034.

[47] Roselin Kiruba, R.—Sree Sharmila, T.: Secure Data Hiding by Fruit Fly Optimization Improved Hybridized Seeker Algorithm. Multidimensional Systems and Signal Processing, Vol. 32, 2021, No. 2, pp. 405–430, doi: 10.1007/s11045-019-00697-w.

[48] Ranganath, H. S.—Kuntimad, G.: Image Segmentation Using Pulse Coupled Neural Networks. Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN '94), Vol. 2, 1994, pp. 1285–1290, doi: 10.1109/ICNN.1994.374369.

[49] Ranganath, H. S.—Kuntimad, G.—Johnson, J. L.: Pulse Coupled Neural Networks for Image Processing. Proceedings IEEE Southeastcon '95: Visualize the Future, 1995, pp. 37–43, doi: 10.1109/SECON.1995.513053.

[50] Forgac, R.—Mokris, I.: Linking and Activation Potential Optimization in the Pulse Coupled Neural Network. 2008 IEEE International Conference on Computational Cybernetics, Stara Lesna, Slovakia, 2008, pp. 85–88, doi: 10.1109/ICCCYB.2008.4721384.

[51] Forgac, R.—Mokris, I.: Feature Generation Improving by Optimized PCNN. 2008 6th International Symposium on Applied Machine Intelligence and Informatics, Herlany, Slovakia, 2008, pp. 203–207, doi: 10.1109/SAMI.2008.4469166.

[52] Forgac, R.—Mokris, I.: Threshold Potential Optimization in the Pulse Coupled Neural Network. 2008 6th International Symposium on Intelligent Systems and Informatics, Subotica, Serbia, 2008, pp. 1–4, doi: 10.1109/SISY.2008.4664914.

[53] Forgáč, R.—Mokriš, I.: Algorithm for Pulse Coupled Neural Network Parameters Estimation. 2009 IEEE International Conference on Computational Cybernetics (ICCC), Palma de Mallorca, Spain, 2009, pp. 147–151, doi: 10.1109/ICCCYB.2009.5393944.

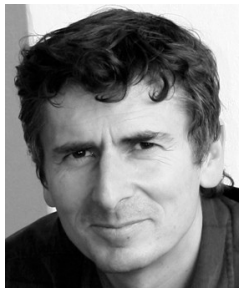
[54] Forgáč, R.—Krakovský, R.: Contribution to Image Steganography Using Pulse Coupled Neural Networks. 2017 Communication and Information Technologies (KIT), Vysoke Tatry, Slovakia, 2017, pp. 1–6, doi: 10.23919/KIT.2017.8109445.

[55] Forgáč, R.—Očkay, M.—Krakovský, R.: Impact of Pulse Coupled Neural Network Parameters on Image Steganography. 2019 Communication and Information Technologies (KIT), Vysoke Tatry, Slovakia, 2019, pp. 1–6, doi: 10.23919/KIT.2019.8883304.

[56] Forgáč, R.—Očkay, M.—Krakovský, R.: Entropy Based Image Quality Assessment of Stego Images Created by Pulse Coupled Neural Network. 2020 New

Trends in Signal Processing (NTSP), Demanovska dolina, Slovakia, 2020, pp. 1–5, doi: 10.1109/NTSP49686.2020.9229546.

[57] Forgáč, R.—Očkay, M.—Javurek, M.: Steganography Based Approach to Image Authentication. 2021 Communication and Information Technologies (KIT), Vysoke Tatry, Slovakia, 2021, pp. 1–6, doi: 10.1109/KIT52904.2021.9583618.



Radoslav Forgáč is a researcher at the Institute of Informatics, Slovak Academy of Sciences. He graduated from the Armed Forces Academy in Liptovský Mikuláš in 1993. He received his Ph.D. in artificial intelligence in 2006 from the Technical University of Košice. His research is focusing on machine learning, especially neural networks for classification, regression and clustering.

Miloš Očkay is a researcher at the Institute of Informatics, Slovak Academy of Sciences and Associate Professor at the Department of Informatics at the Armed Forces Academy in Liptovský Mikuláš. He holds his Ph.D. degree in the field of informatics, received in 2012 from the Technical University of Košice. His research is focusing on parallel computing, neural networks.

Martin Javurek is an Assistant Professor at the Department of Informatics at the Armed Forces Academy in Liptovský Mikuláš and a researcher at the Institute of Informatics, Slovak Academy of Sciences. He holds his Ph.D. degree in the field of informatics, received in 2017 from the Armed Forces Academy in Liptovský Mikuláš. His research focuses on neural networks, cybersecurity and cryptography.

Bianca Badidová is a Ph.D. candidate at the Department of Informatics at the Armed Forces Academy of Liptovský Mikuláš. Her research focuses on data analysis, cybersecurity and neural networks.

Computing and Informatics, Vol. 42, 2023, 615–650, doi: 10.31577/cai_2023_3_615

ATTRIBUTE-BASED ACCESS CONTROL POLICY GENERATION APPROACH FROM ACCESS LOGS BASED ON THE CATBOOST

Shan Quan, Yongdan Zhao, Nurmamat Helil*

College of Mathematics and System Science
Xinjiang University
China
e-mail: shanquan_owen@163.com, zydky2021@126.com, nur924@sina.com

Abstract. Attribute-based access control (ABAC) has higher flexibility and better scalability than traditional access control and can be used for fine-grained access control of large-scale information systems. Although ABAC can depict a dynamic, complex access control policy, it is costly, tedious, and error-prone to manually define. Therefore, it is worth studying how to construct an ABAC policy efficiently and accurately. This paper proposes an ABAC policy generation approach based on the CatBoost algorithm to automatically learn policies from historical access logs. First, we perform a weighted reconstruction of the attributes for the policy to be mined. Second, we provide an ABAC rule extraction algorithm, rule pruning algorithm, and rule optimization algorithm, among which the rule pruning and rule optimization algorithms are used to improve the accuracy of the generated policies. In addition, we present a new policy quality indicator to measure the accuracy and simplicity of the generated policies. Finally, the results of an experiment conducted to validate the approach verify its feasibility and effectiveness.

Keywords: ABAC policy, access logs, policy mining, ensemble learning, CatBoost

1 INTRODUCTION

In the big data era, big data platforms can help the information systems of organizations and enterprises overcome data isolation; support the integration of multi-

* Corresponding author

source heterogeneous data; and support cross-industry, cross-department, and cross-platform data sharing and exchange. As a result, data are now among the most strategic assets of any government, organization, or enterprise. Dengguo et al. [1] defined big data as the process of obtaining useful knowledge and predicting future trends, analyzing and grasping data's essential characteristics, and using the results of the analysis to distinguish the true from the false. Undoubtedly, big data has enduring value. However, it comes with the problem of data security. Ensuring that unauthorized entities do not access data is a security problem that must be solved in the process of data use. Access control is an essential solution to this problem.

In the big data environment, the access control system has many subjects, objects, and dynamic changes. Data structures and sources are complex and diverse. User types, demands for information sharing, and privacy needs are great. Moreover, access permissions are constantly changing [2]. Early access control models such as discretionary access control (DAC) [3, 4] and mandatory access control (MAC) [5] are not very suitable for addressing access control policies in the big data environment. The role-based access control (RBAC) model [6] maps users to roles through which they possess permissions. As the basis of the modern access control model, RBAC has been one of the popular research areas in access control, but RBAC relies heavily on user identity. The attribute-based access control (ABAC) model [7] later emerged as a fine-grained access control mechanism that relies on attributes. ABAC solves the problems of expressing and enforcing fine-grained access control and large-scale user dynamic expansion in a complex information system. Moreover, ABAC embeds entity attributes into the access control policy (ACP). As the subject, object, environment, and operation attributes have the ability to describe the access control and constraints in ABAC, the model has sufficient flexibility and extensibility.

With the rapid development of cloud computing, big data, artificial intelligence, and other technologies, the number of entities in information systems has exploded. ABAC uses subject and object attributes as essential criteria for permission access. The introduction of the environment attributes enables ABAC to support dynamic access control [8, 9, 10]. Unlike RBAC, ABAC does not need to design complex roles in advance, thus effectively avoiding the role explosion in RBAC [7]. However, when the number of subjects and objects and the number of subject and object attributes become large, it is challenging to specify ABAC policy manually. This is time-consuming and expensive, which makes ABAC's deployment in practical applications difficult [10]. Therefore, the research on ABAC policy mining is of great significance and can promote the development and popularization of the ABAC model.

As it is challenging to define ABAC policy manually, this paper proposes an approach to ABAC policy generation from access logs based on the CatBoost algorithm. This is an integrated learning method that can automatically learn ABAC policy from historical access logs. First, we reconstruct the attributes of the policies that need to be mined by weighting. Subsequently, we propose the rule extraction algorithm, rule pruning algorithm, and rule optimization algorithm to improve the

accuracy of the generated policy. In addition, we propose a new policy quality indicator, namely the policy quality comprehensive indicator, which measures the accuracy and conciseness of the generated policy.

The rest of this article is organized as follows. In Section 2, we review the research background and related work. In Section 3, we summarize the prior knowledge of ABAC and the CatBoost machine learning (ML) algorithm and detail the preparatory work of ABAC policy generation. In Section 4, we introduce the generation process of the ABAC policy and give the related implementation algorithm in detail. In Section 5, we provide the experimental results and evaluation. Finally, in Section 6, we present the conclusion and research prospects.

2 RELATED WORK

The ABAC model is widely used in large distributed environments, web service systems, grid computing, and information sharing and management [1]. In ABAC deployment, one of the critical challenges is how to infer ACP from the logs of past decisions (*permit* or *deny*) on the access requests made by users. In the ABAC access control system, two primary sources of information describe the relations between subjects and objects: the original access control system and the access logs [11]. The basic idea of policy mining is to combine subject, object, environment, and operation attribute data to mine ABAC policies from the relations between the subject and object. Therefore, mining ABAC policies from access logs has attracted the attention of researchers.

The initial research field of policy mining was RBAC role mining. Vaidya et al. [12] introduced an approach of role mining that finds the best role from the user-permission assignment relations by decomposing the user-permission Boolean matrix. Molloy et al. [13] proposed an RBAC role mining algorithm based on formal concepts. Molloy et al. [14] proposed a role mining approach that allows noisy data. Most role mining approaches assume that the data used are correct and noise-free, which is often not the case. Thus, this approach improves the quality of role mining. Currey et al. [15] proposed a multi-objective role mining approach that minimizes unnecessary permissions as the formal goal of role mining. Jafarian et al. [16] transformed the role mining problem into a constraint satisfaction problem. This approach effectively combines top-down and bottom-up patterns. The top-down pattern starts from the security requirements and then gradually refines the business and then dissolves into independent functional units to generate policies. The bottom-up pattern starts with access requests and uses the common ground among access requests to generate policies. Combining the two patterns makes the generated policies easier to understand and maintain and of higher quality.

Some scholars have proposed RBAC policy mining approaches based on ML. Molloy and Chari [17] proposed an RBAC role mining approach with permissions, which is based on ML algorithms. Their approach has advantages in generality, coverage, and stability. Narouei and Takabi [18, 19] proposed an approach of uti-

lizing natural language processing (NLP) technology called semantic role labeling (SRL), which extracts ACPs from unrestricted natural language documents, defines roles, and constructs an RBAC model. It is a top-down pattern for role mining. As this approach considers all predicates in the ACPs sentence, it leads to some false positives, which makes their approach's precision relatively low (precision of 75 %). Anderer et al. [20] created a library of role mining benchmark instances, which includes some new, synthetically generated benchmark instances of different sizes for evaluating and comparing role mining algorithms. The benchmark instances leave more space between the number of roles derived from the two common decompositions of the role mining problem (RMP) and the actual minimum number of roles, thus making them better, multifaceted, and able to thoroughly evaluate the role mining algorithm.

As ABAC is widely used in access control, researchers have also proposed ABAC policy mining approaches. Chari and Molloy [21] proposed mining ABAC rules automatically from access logs instead of manually making and maintaining the ABAC rule set. They used cross entropy to exclude user attributes to mine a rule set. Xu and Stoller proposed ABAC policy mining algorithms from access logs [11], RBAC [22], and the access control list (ACL) and attribute data [23]. Their algorithms iterate over access control tuples and build candidate rules, and then generalize them by replacing the conjunctions in the attribute expressions with constraints. Iyer and Masoumzadeh [24] proposed an approach to mine positive and negative ABAC policies. It can extract (*permit* or *deny*) the ACP at the same time. The mining policy is also relatively concise, thus making it superior to a previous approach [23]. Chakraborty et al. [25] defined the existence problem of the ABAC rule set and provided an algorithm to solve it. They further introduced the concept of the infeasible rule set modification in ABAC and the modification algorithm. Talukdar et al. [26] proposed an algorithm that finds the most general rule from a set of candidate rules, which can automatically build a reasonable ABAC policy. The main advantage of this approach is that the running time is stable and is not affected by the number of attributes. Narouei et al. [27] proposed an approach of ABAC policy mining based on the particle swarm optimization algorithm and ABAC policy mining under the minimal perturbation problem, and proposed a global optimization function to obtain the optimal ABAC state while making it as similar as possible to the existing state. Medvet et al. [28] proposed an evolutionary approach of multi-objective strategy mining based on genetic operators. It generates strategies through iterative, evolutionary search. Each iteration learns new rules and makes the set of access control tuples smaller to improve the quality of mining rules. Das et al. [29] proposed an ABAC policy mining algorithm based on the Gini coefficient impurity. The algorithm considers the environment attributes and their associated values and uses the approach based on the decision tree (DT) to build the policy. Although the generated rules are few and compact, access control decisions can be made faster.

Owing to the rapid development of big data, artificial intelligence and other related technologies (e.g., ML) have been widely used. As a result, some researchers

have proposed ABAC policy mining algorithms based on ML. Cotrini et al. [30] proposed an algorithm named Rhapsody to mine ABAC rules from sparse logs. They also defined the concept of reliability to measure the reliability of the extracted rules. The algorithm also considers whether the generated policies are overly permissive. Karimi and Joshi [31] proposed an approach that uses unsupervised learning to detect specific patterns in a set of access records and then extract ABAC policies from these patterns. In addition, they provided two algorithms, rule pruning and policy refinement, which are used to improve policy quality. Das et al. [32] provided a visual ABAC policy mining approach. It represents the existing access requests in the form of a binary matrix and then transforms the problem of finding the best representation of the binary matrix into a minimization problem by extracting rules from the visual access control matrix.

Some scholars have proposed ABAC policy mining approaches based on neural network (NN) and reinforcement learning (RL). Narouei et al. [33, 34] provided an information extraction approach from natural language documents via a recurrent neural network (RNN) and SRL. It can identify access control policy statements, and its performance was 5.58 % higher than that of the support vector machine model. Alohaly et al. [35, 36] proposed a convolutional neural network attribute extraction approach. Their approach *F1-score* performs well; it can generate a practical framework for analyzing natural language access control policies; and it can identify the attributes of the subjects and object elements. Karimi et al. [37] proposed an adaptive ABAC policy learning approach that can realize the automation of decisions. It is a kind of RL. This approach shows good performance in policy transfer using the learning feedback mechanism, and it is superior to the approach based on supervised learning.

Here, we focus on reviewing the approaches of [38] and [39]. The [38] and [39] use restricted Boltzmann machine (RBM) and multi-layer perceptron (MLP) to mine ABAC policies, respectively. Mocanu et al. [38] presented an ABAC policy mining approach based on deep learning that uses the RBM algorithm to train logs and extract rules. They first summarize knowledge from logs and generate a set of candidate rules in binary vector format and then convert the candidate rule set from the binary vector format to an acceptable format. Finally, the reconstruction error of all log entries in the obtained model is calculated, and the maximum value is taken as the threshold. Then, they generate all possible rule combinations, calculate the reconstruction error in the obtained model, and add the rules whose reconstruction error is less than the threshold to the candidate rules. The implicit distribution of data can be found through this approach. This approach has strong anti-noise ability, but it does not further optimize and analyze the rule set. Cappelletti et al. [39] deduced ABAC policy by comparing different symbolic and non-symbolic ML techniques. They used MLP to infer ABAC policies from access logs and turn them into a classification problem. MLP is a neural network model, which maps multiple input datasets to a single output dataset. It provides a deep feed-forward artificial network and generates a set of outputs from one set of inputs and the other end. Its feature is that several layers of input nodes are connected as a directed graph

between the input and output layers. MLP had a relatively good policy decision result in the experiment compared with other approaches.

ABAC is the most prominent access control model. Therefore, to improve the efficiency and accuracy of access request decisions, ABAC policy mining is a topic worth studying. As logs reflect the ACPs and user behaviors implemented in an organization, mining ACPs or rules from logs can help us reconstruct and simplify complex and dynamic policies. Researchers have successively proposed policy mining algorithms based on Rhapsody, unsupervised learning, neural networks, and RL. Rhapsody, unsupervised learning, NNs, RL, and other ML-based policy mining algorithms are more practical and effective, but they have some deficiencies. For example, in these algorithms, the decision is overly permissive, and only policy attributes are extracted. Although these algorithms improve decision efficiency in various ways, in practice, there are still some problems that require further study, such as no optimized rules and poor accuracy.

3 PRELIMINARIES FOR ABAC POLICY MINING

This paper uses the ML algorithm based on CatBoost to study how to mine more accurate and reasonable policies. The CatBoost algorithm is primarily used for classification, prediction, and regression. CatBoost has a wide range of application scenarios and can deal with gradient bias and prediction shift, which improves the accuracy and generalization ability of the algorithm. For the ABAC policy mining problem, we choose the CatBoost algorithm mainly because of its practicality, robustness, accuracy, and extensibility. This paper makes the following contributions:

1. We propose an ABAC policy generation method based on CatBoost, in which we learn ABAC policy from historical access logs.
2. We perform a weighted reconstruction of the attributes for the ABAC policy to be mined, which helps improve the accuracy and rationality of ABAC rule extraction.
3. We propose a rule extraction algorithm, rule pruning algorithm, and rule optimization algorithm to improve the accuracy of the generated policies.
4. We propose a new policy quality indicator, namely the policy quality comprehensive indicator, to measure the accuracy and simplicity of the generated policies.

In this section, we give a brief overview of ABAC, data mining (DM), the CatBoost ML algorithm, and some preparations for ABAC policy mining.

In this article, we try to follow the National Institute of Standards and Technology ABAC standard [7]. We use user attributes, object attributes, and session attributes to refer to access requester attributes, object attributes, and environment attributes (or conditions), respectively.

3.1 Learning CatBoost

The full name of CatBoost is Categorical Boosting. This algorithm is a machine learning algorithm proposed by the Russian search giant Yandex in 2017. It belongs to the boosting family of algorithms in integrated learning. It is an integrated algorithm combining gradient boosting and Oblivious Trees [40, 41, 42]. It is suitable for heterogeneous (different types) data and can handle gradient deviation and prediction deviation problems. Therefore, this algorithm can not only improve the quality and prediction speed of the classification model but also significantly improve the accuracy and generalization ability of the algorithm. The main advantages of the CatBoost algorithm are as follows [40, 41, 42]:

1. Practicability: It can process categorical and numerical data and it supports categorical variables without requiring preprocessing of the non-numerical features;
2. Robustness: High-quality models can be obtained without parameter adjustment, and very good results can be obtained by using the default parameters, thus reducing the time spent on parameter adjustment and the need for superparameter tuning;
3. Accuracy: It uses a new gradient lifting algorithm to build the model, thus reducing overfitting and improving the accuracy of the model;
4. Extensibility: It supports user-customized loss functions.

3.2 Preliminaries

Generally, when processing attributes in data, it is necessary to determine the attribute types beforehand, as the chosen processing methods differ against different types of attributes. Attributes are used to describe the properties or characteristics of an entity that vary from entity to entity and change over time (e.g., teachers' job titles, students' ages, and courses). In general, attributes can be divided into five categories: ordinal, nominal, interval, ratio [43], and binary. A binary attribute has only two states, denoted by `true` and `false` or 0 and 1, respectively.

Nominal, binary, and ordinal attributes are called categorical attributes, which are qualitative and discrete. Interval and ratio attributes are also called numeric attributes. A numeric attribute is quantitative, and its value can be discrete or continuous. The attributes' specific descriptions are shown in Table 1.

Let $En = U \cup O \cup E$ be the set of all entities and $A = A_u \cup A_o \cup A_e$ be the set of all attributes in the ABAC system. Here, U, O, and E are the ABAC system's sets of users (subjects), objects, and environments, respectively. Each element (entity) in U, O, and E is expressed by the Boolean combination of related attributes. In addition, OP is a set of operations in the system. A_u, A_o, and A_e are the set of user (subject) attributes, object attributes, and environment attributes, respectively.

Attribute Types		Attribute Description	Attribute Example	Attribute Analysis
Categorical attribute	Ordinal attribute	Attribute values have an order or size	Education, grade etc.	Median, percentile etc.
	Nominal attribute	Represents the category, code, or status	Native place, name, occupation, etc.	Mode, entropy, etc.
	Binary attribute	With only two categories or states	Flip a coin for positive or negative, nucleic acid test results for positive or negative, etc.	Contingency correlation etc.
Numerical attribute	Interval attribute	Comparing the difference is significant	Temperature, time, date, etc.	Mean, standard deviation, etc.
	Ratio attribute	Calculating the ratio or difference is necessary	Age, length, percentage of project completed, etc.	Geometric mean, harmonic mean, etc.

Table 1. Classification of attributes

Definition 1 (Attribute Domain)**.** Let the attribute $a \in A$. The set of all valid values of a is called the attribute domain of a, denoted as $V(a)$.

Definition 2 (Attribute Relation)**.** Define the binary relation $F = \{\langle a, v\rangle \mid a \in a, v \in V(a)\}$ as an attribute relation.

Definition 3 (Access Request)**.** The access request (ar) is a four-tuple $ar = (u, o, e, op)$, which is explained as follows: User $u \in U$ sends an access request to the system, requesting that it perform the operation $op \in OP$ on the object $o \in O$ under the environmental condition $e \in E$. u, e, o are determined by specific attribute relations.

Definition 4 (Access Control Decision)**.** An access control decision is a five-tuple $acd = (ar, d) = (u, o, e, op, d)$, composed of a user, an object, the environment, an operation, and the decision. Here, $d \in \{permit, deny\}$.

An access control decision result is either *permit* or *deny*. When the decision is *permit*, the user (requester) can perform the given operation on the given object under the given environment. When the decision is *deny*, the user cannot perform the given operation on the given object under the given environment.

Definition 5 (Access Logs)**.** The access logs (L) are a set of access control decisions.

As the decision result is *permit* or *deny*, we can divide the access logs (L) into positive access logs (L^+) and negative access logs (L^-). That is,

- $L^+ = \{(ar, d) \mid d = permit\}$,
- $L^- = \{(ar, d) \mid d = deny\}$.

Definition 6 (Access Rule)**.** An access rule refers to a multi-tuple $r = (F, op, d)$, where F is an attribute relation that includes relations regarding users, objects, and environments. It can be written as $F = F_u \cup F_o \cup F_e$. op is an operation, and d is a decision.

Example 1. A rule $r = (\{\langle Position, Student\rangle, \langle Location, Campus\rangle, \langle Type, Book\rangle, \langle Id_{library}, Id_{student}\rangle\}, borrow, permit)$ can be explained as "if a student is on campus and his/her student number matches the library code, he/she is permitted to borrow a book from the library."

Definition 7. Given the four-tuple $t = (u, o, e, op)$ from an access request $ar = (u, o, e, op)$ or an access control decision $acd = (u, o, e, op, d)$, and the rule $r = (F_r, op_r, d_r)$, if $F_u \cup F_o \cup F_e \subseteq F_r$, where F_u, F_o and F_e is the attribute relation of user u, object o, and environment e in the access request or the access control decision, $op = op_r$; then, we say the four-tuple satisfies rule r, denoted as $t \models r$. For simplicity, we say the access request ar (access control decision acd) satisfies rule r, denoted as $ar \models r$ ($acd \models r$).

In Definition 7, we mainly consider the satisfiability between the four-tuple (u, o, e, op), from an access request ar (an access control decision acd) and a rule.

Thus, regarding the rule set R in an ABAC system,

$$R \subseteq F_U \times F_O \times F_E \times OP \times D,$$

where F_U, F_O, and F_E are the set of attribute relations of all users in U, objects in O, and environments in E, respectively; $D = \{permit, deny\}$.

Definition 8. Permission $pe = (o, e, op)$ is defined as the operation of a user (subject) on an object under an environment, which is expressed by an object, the environment-related attribute relations, and an operation.

Definition 9 (ABAC Instance)**.** An ABAC instance is a subset of the multivariate relation $AR \times d$, denoted as I_{AR}, where AR represents the set of access requests (ar), and d is the set of decision results (*permit* or *deny*). I^+_{AR} and I^-_{AR} are defined as subsets of I_{AR}, where the decision in this instance is *permit* or *deny*, respectively.

Position \ Location	Campus		Home			
Professor	√	√	√	•	•	√
	√	√	√	•	√	•
Associate Professor	√	√	×	•	•	•
	√	√	•	•	•	×
Lecturer	√	√	•	•	×	•
	√	•	•	×	•	•
Student	•	•	•	•	×	×
	•	•	×	•	•	•

Table 2. An instance of the access log. Each tick √, cross ×, and circle • denotes an access request (i.e., a user). The ticks √ and crosses × denote logged requests that have been permitted and denied, respectively. The circles • denote users who have not requested permission yet.

Definition 10 (Rule Confidence)**.** Let $AR \times d$ be an instance of ABAC; then, the confidence of rule r is defined as follows:

$$Conf(r) = \frac{|I_{AR}^{+}|}{|I_{AR}|}, \tag{1}$$

where I_{AR} represents the set of all requests in the instance that satisfy rule r, and I_{AR}^{+} denotes the set of all permitted requests in the instance that satisfy rule r. If $|I_{AR}| = 0$; then, we define $Conf(r) = 0$.

Example 2. In Table 2, the confidence of rules $r_1 = (\{\langle Location, Campus\rangle\}, borrow, permit)$ is $\frac{11}{16} \approx 0.69$, and the confidence of rules $r_2 = (\{\langle Position, Professor\rangle\}, borrow, permit)$ is $\frac{8}{12} \approx 0.67$.

Definition 11 (Rule Refinement)**.** Given two rules $r = (F, op, d)$ and $r' = (F', op', d')$, if $F \subset F'$ ($F_u \subset F'_u \wedge F_o \subset F'_o \wedge F_e \subset F'_e$), $op = op'$, $d = d'$, then r' adds new constraints on r. r' is called refinement of r, the refinement relation is denoted as $r \propto r'$.

Definition 12. For the given refinement relation $r \propto r'$, we say rule r' is overly permissive if $Conf(r') < Conf(r)$.

Example 3. As shown in Table 2, consider two refinements of the rule $r_1 = (\{\langle Location, Campus\rangle\}, borrow, permit)$:

- $r_{11} = (\{\langle Location, Campus\rangle, \langle Position, Professor\rangle\}, borrow, permit)$;
- $r_{14} = (\{\langle Location, Campus\rangle, \langle Position, Student\rangle\}, borrow, permit)$.

These refinements have confidence 1.0 and 0, respectively.

As shown in Example 3, we can see that the rule $r_{14} = (\{\langle Location, Campus\rangle, \langle Position, Student\rangle\}, borrow, permit)$, and its confidence is decreased to 0; so, this rule is overly permissive.

Definition 13 (Rule Credibility)**.** Let $AR \times d$ be an instance of ABAC; the credibility of rule r is defined as

$$Cre_T(r) = \min_{r' \in F_T(r)} \{Conf(r), Conf(r')\}, \tag{2}$$

where $F_T(r) = \{r' \mid |[r']| \geqslant T\}$, $[r']$ is the set of refinements of r, and T is a parameter specified by a policy administrator. Generally, the optimal value of T is the minimum number of times the refinement r' satisfies an access request. If $F_T(r) = 0$, we define $Cre_T(r) = Conf(r)$.

Example 4. We compute the rule credibility for the rules $r_1 = (\{\langle Location, Campus\rangle\}, borrow, permit)$ and $r_2 = (\{\langle Position, Professor\rangle\}, borrow, permit)$ for the instance of Table 2.

Consider the below refinements of rule $r_1 = (\{\langle Location, Campus\rangle\}, borrow, permit)$:

- $r_{11} = (\{\langle Location, Campus\rangle, \langle Position, Professor\rangle\}, borrow, permit)$;
- $r_{12} = (\{\langle Location, Campus\rangle, \langle Position, AssociateProfessor\rangle\}, borrow, permit)$;
- $r_{13} = (\{\langle Location, Campus\rangle, \langle Position, Lecturer\rangle\}, borrow, permit)$;
- $r_{14} = (\{\langle Location, Campus\rangle, \langle Position, Student\rangle\}, borrow, permit)$.

$Cre_4(Conf(r_1)) = \min\{Conf(r_1), Conf(r_{11}), Conf(r_{12}), Conf(r_{13}), Conf(r_{14})\} = \min\{0.69, 1.0, 1.0, 0.75, 0.0\} = 0$.

Consider the below refinements of rule $r_2 = (\{\langle Position, Professor\rangle\}, borrow, permit)$:

- $r_{21} = (\{\langle Position, Professor\rangle, \langle Location, Campus\rangle\}, borrow, permit)$;
- $r_{22} = (\{\langle Position, Professor\rangle, \langle Location, Home\rangle\}, borrow, permit)$.

$Cre_2(Conf(r_2)) = \min\{Conf(r_2), Conf(r_{21}), Conf(r_{22})\} = \min\{0.67, 1.0, 0.5\} = 0.5$.

Theorem 1. Let $T \geqslant 1$, $K \in [0, 1]$; r is a rule, K is a specified value, and in general, $K \approx \frac{|AR^+|}{|AR|}$, AR^+ is the set of permitted access requests. If there is a refinement relation $r \propto r'$ that satisfies $Conf(r') < K$ and $|[r']| \geqslant T$, then r' is overly permissive.

Proof. From the Definitions 10, 12, and 13, we can see that if a rule r' ($r \propto r'$) is overly permissive, then its confidence must be less than the confidence of the original rule (that is, equal to K), so $Conf(r') < K$ and $|[r']| \geqslant T$. If the refinement r' satisfies $Conf(r') < K$ and $|[r']| \geqslant T$, then the refinement r' is overly permissive. □

Corollary 1. If $Conf(r') \geq K$, $|[r']| \geqslant T$, and r' is not overly permissive, we have credibility $Cre_T(r') \geq K$.

Proof. From the Definitions 10, 12, and 13, similarly, Corollary 1 is also true. □

Definition 14 (Log Credibility)**.** The log credibility is defined as

$$Cre_T(r) = \min_{r' \in F_T(r)} \{Conf(r), Conf(r')\}. \tag{3}$$

However, in the logs, the confidence cannot be directly computed in the instance; so, we denote the set of records in the access log set that meets rule r as $\{r\}_L^I$ $(I \subseteq L)$. In the logs, we have

$$F_T(r) = \{r' \mid |[r']| \geqslant T\}, \tag{4}$$

$$Conf(r) = \frac{|\{r\}_L^{L^+}|}{|\{r\}_L^L|}, \tag{5}$$

where L^+ indicates the positive (*permit*) access logs.

3.3 Policy Generation-Related Methods and Evaluation Metrics

In practice, it is difficult to see the correlation between the features and targets and the correlation between the features. Therefore, using mathematical or engineering methods is necessary to help improve feature selection. This paper uses an embedded method to select the features. For details, see Figure 1.

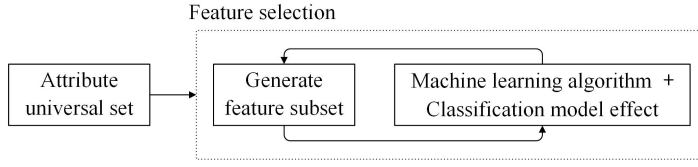


Figure 1. Embedded method for feature selection

The mutual information between the random variables X and Y is the mathematical expectation of mutual information between individual events, which is also used to evaluate the correlation between variables. The mutual information calculation formula is as follows:

$$I(X;Y) = E[I(x_i;y_i)] = \sum_{x_i \in X} \sum_{y_i \in Y} p(x_i, y_i) \log \frac{p(x_i, y_i)}{p(x_i)p(y_i)}, \tag{6}$$

$$I(X;Y) = H(X) - H(X \mid Y) = -\sum_{x_i \in X} p(x_i) \log p(x_i) - \left(-\sum_{x_i \in X, y_i \in Y} p(x_i, y_i) \log p(x_i \mid y_i) \right). \tag{7}$$

We can evaluate the CatBoost ML algorithm by k-fold cross-validation according to accuracy and other indicators. The evaluation is carried out on the cross-validation dataset. Based on this, the following definitions are given.

- *True Positive* (TP): If the access request is permitted based on the actual policy, it is also permitted based on the generated policy;
- *True Negative* (TN): If the access request is denied based on the actual policy, it is also denied based on the generated policy;
- *False Positive* (FP): If the access request is denied based on the actual policy, it is permitted based on the generated policy;
- *False Negative* (FN): If the access request is permitted based on the actual policy, it is denied based on the generated policy.

We can obtain *True Positive Rate* (TPR), *True Negative Rate* (TNR), *False Positive Rate* (FPR), *False Negative Rate* (FNR), *Accuracy* (Acc), *Sensitivity/Recall*, *Precision*, *F1-score* and *Matthews correlation coefficient* (Mcc), through the above definition. The calculation formulas are as follows:

$$TPR = Sensitivity = Recall = \frac{TP}{TP + FN}, \tag{8}$$

$$TNR = Specificity = \frac{TN}{TN + FP}, \tag{9}$$

$$FPR = \frac{FP}{FP + TN}, \tag{10}$$

$$FNR = \frac{FN}{FN + TP}, \tag{11}$$

$$Precision = \frac{TP}{TP + FP}, \tag{12}$$

$$Accuracy(Acc) = \frac{TP + TN}{TP + TN + FP + FN}, \tag{13}$$

$$F1\text{-}score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{14}$$

$$Mcc = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}. \tag{15}$$

The increase in ACPs makes maintenance more difficult and increases the computational overhead. So, complexity and accuracy are essential evaluation indicators of policy quality. Therefore, weighted structural complexity (WSC) and accuracy are considered in this paper. In addition, the influence of other evaluation indicators on policy quality is considered.

WSC is used to summarize the size of the policy. Molloy et al. introduced *WSC* [13] into the artificial mining of RBAC policy. Later, Xu and Stoller extended it to mining ABAC policy [23]. The simpler the policy, the easier it is to manage in the system. *WSC* is the weighted sum of the number of elements of the ABAC policy with regard to π. Its calculation formula is as follows:

$$WSC_\pi = \sum_{r \in \pi} WSC(r), \tag{16}$$

$$WSC(r) = w_1 \cdot WSC(F_u) + w_2 \cdot WSC(F_o) + w_3 \cdot WSC(F_e) + w_4 \cdot WSC(OP). \tag{17}$$

For $\forall\ F_u,\ F_o,\ F_e,\ OP,\ WSC(F_u) = |F_u|,\ WSC(F_o) = |F_o|,\ WSC(F_e) = |F_e|,\ WSC(OP) = |OP|$, where $w_{i \in \{1,2,3,4\}}$ is the specified weight.

3.4 ABAC Policy Mining

In this section, we discuss ABAC policy mining (ABAC-PM) based on the characteristics of ABAC and the factors and challenges that need to be considered in mining ABAC policies.

Definition 15 (ABAC Policy)**.**[1] An ABAC policy (π) is a set of rules with the same permission. That is,

$$\pi = \left\{ \bigvee_{i}^{n} r_i \mid r_i \in R \wedge F_{o_{r_i}} = F_{o_{r_j}}, F_{e_{r_i}} = F_{e_{r_j}}, op_{r_i} = op_{r_j}, \right.$$
$$\left. d_{r_i} = d_{r_j}, \forall i, j \in \mathbb{N}^*, i \neq j \right\},$$

where $F_{o_{r_i}}$, $F_{o_{r_j}}$, $F_{e_{r_i}}$, $F_{e_{r_j}}$, op_{r_i}, op_{r_j}, d_{r_i}, d_{r_j} represent the subject, object, and environmental attributes relation, operation and decision of two different rules, respectively.

ABAC Policy Mining. If given a set of subjects (S), a set of objects (O), a set of environments (E), a set of operations (OP), and attribute domains and access logs or an ACL, we need to construct an ABAC policy set (Π). We have the following requirements for policy mining:

1. Every access request in the access logs or ACL satisfies at least one rule in policy $\pi \in \Pi$;
2. For any rule $r \in \pi$, it is as concise as possible;
3. The number of rules in π is as small as possible, and the accuracy is as high as possible.

[1] The basic unit of a policy in ABAC is a rule. We do not strictly differentiate between rule and policy in this article.

Suppose a given original access control system has the access control decisions of *permit* and *deny* against requests; it is related to numerous attributes; and the collected data is complex. In that case, it is ideal to use the CatBoost algorithm to mine the ACP and write it as a form of the ABAC policy. The ABAC policy extraction can be regarded as establishing a mapping between the access control decision data, including the user, object, environment attributes, and the set of ABAC policies. This mapping can be represented by the function $LF : L \to Y$, where

1. L represents a set of access control decisions (access logs): The components of each tuple in this work are restricted to categorical or nominal variables (e.g., the category value of an attribute);
2. Y represents a set of numbered labels (set labels), each corresponding to a rule in the ABAC policy π.

We aim to make the loss function (LF) (i.e., the function of classification error in machine learning) smaller and to mine the desired policy with high precision and efficiency.

4 ABAC POLICY GENERATION

Manually defining ABAC policies is expensive and time-consuming; so, an automated approach to mining ABAC policies helps simplify the adoption or migration of ABAC policies. Therefore, this section discusses the ABAC policy generation method based on CatBoost.

4.1 ABAC Policy Extraction

The ABAC policy extraction problem essentially involves finding rule r from a log dataset (L), making the ABAC policy (π) concise and more accurate when making decisions. We use CatBoost to extract the ABAC policy. The specific process of policy extraction is shown in Figure 2, including implementation steps and a summary.

4.2 Access Logs Preprocessing

After collecting the access logs, we first preprocess them to map the attribute values of the subject, object, and environment attributes to a type value. Some attribute values may be missing in the access logs; so, missing attribute values are also handled in this step. In the classification process, missing attribute values are usually replaced by the most common ones. However, the policy extraction approach in this paper is sensitive to the occurrence frequency of each attribute value. Thus, if an attribute is a valid attribute, its missing value is replaced by UNK (unknown) in the corresponding data.

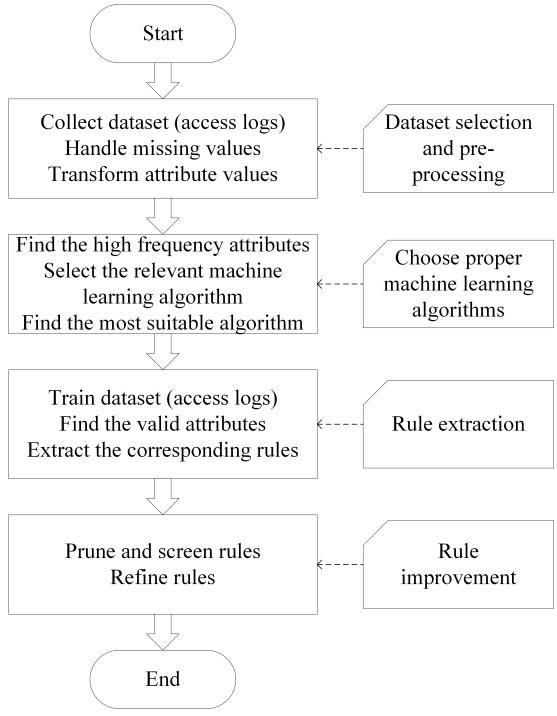


Figure 2. Rule generation process

4.3 Attribute Selection

Owing to the diversity and differences of attributes, the categorical attributes of some entities need to be encoded before attribute selection, primarily by using ordinal encoding and one-hot encoding. We use the recursive feature elimination approach based on CatBoost to select the attributes.

The access log records the attribute information of the subject, object, and environment and the result of the decision operation. First, the attribute combination is transformed into feature vectors. Then, we treat these feature vectors as training data. The decision result of *permit* and *deny* becomes the label of the training data. The model is trained by the CatBoost algorithm, and the trained classification accuracy is regarded as a critical evaluation indicator for attribute deletion. If an attribute is more important, the deletion of the attribute has a more significant impact on the accuracy of the classification algorithm, and its deletion reduces the accuracy of the classification. On the contrary, if an attribute is less important, deleting the attribute has less influence on the accuracy of the classification algorithm.

Then, the scale of the attributes is reduced according to the backward sort method, and the attributes with lower importance are deleted by the iterative method. The Algorithm 1 is a policy attribute selection algorithm based on the greedy elimination algorithm. The final attribute importance sequence is the inverse sequence of the attribute deletion sequence.

Algorithm 1 Attributes selection algorithm

Input: L, A, m // m is a threshold, which is determined by the training model and the access logs

Output: A^*

1: **procedure** SelectAttribute(L, A, m)
2: $A^* \leftarrow \emptyset$
3: $N_a \leftarrow CopyAttributeSet(L, A)$
4: **while** $|N_a| \geq m$ **do**
5: $T_m \leftarrow TrainAccessLogs(L, N_a)$
6: $S \leftarrow SortAttribute(T_m)$
7: $c \leftarrow GetLowerImportanceAttribute(S)$
8: $N_a \leftarrow N_a - \{c\}$
9: **end while**
10: $A^* \leftarrow N_a$
11: **return** A^*
12: **end procedure**

The detailed analysis of the Algorithm 1 is as follows. Line 3, it clones the attribute set. Next, lines 4–9 use the CatBoost algorithm to train the logs and attribute set and sort the attributes; the attributes with lower importance are deleted. Finally, line 10 gets the attributes with higher importance.

4.4 Rule Extraction

Before rule extraction, we propose a method for determining the attribute and operation weight, which is described as follows:

$$AOW = \sum w \cdot A = \sum (w_u \cdot A_u + w_o \cdot A_o + w_e \cdot A_e + w_{op} \cdot OP). \tag{18}$$

Attribute and operation weight (AOW) refers to the weights of the attributes and operations. w_u, w_o, w_e, and w_{op} represent the weight of the user (subject) attributes, object attributes, and environment attributes and operations, respectively. The weights are determined by the training data results of the CatBoost model and are different for different data and models.

Afterward, we need to determine the attribute relation (F), operation set (OP), and the minimum number of times the rule satisfies the access request (T) (Theorem 1) and calculate the confidence degree ($Conf(r)$) and threshold (K) of the rule.

The rule extraction algorithm uses CatBoost to mine frequent sets until no frequent sets can be found.

The Algorithm 2 gives the process of rule extraction in detail.

Algorithm 2 Rule extraction algorithm

Input: A^*, L, D, T, K
Output: R

```
1: procedure ExtractRule(A*, L, D, T, K)
2: R ← ∅
3: F ← GetAttributeRelation(L, A*)
4: OP ← GetOperation(L, D)
5: ar ← GetAccessRequest(F, OP)
6: X ← SaveMatrix(L, F, ar)
7: FreAttrSet ← GetFrequentAttributeSet(X, T)
8: R_c ← GetCandidateRule(FreAttrSet, ar)
9: R_s ← RuleSort(R_c)
10: for all r_i ∈ R_s do
11:     if length(r_i) = 0 then
12:         R_s ← R_s − {r_i}
13:     end if
14: end for
15: for all r_i ∈ R_s do
16:     if Conf(r_i) < K then
17:         R_s ← R_s − {r_i}
18:     end if
19: end for
20: R ← R_s
21: return R
22: end procedure
```

The Algorithm 2 is described as follows. Line 3 gets the attribute relation F. Line 4 gets the corresponding operation set OP. Line 5 gets the corresponding access request ar according to the operation and F. Line 6 finds the F of all access control records in L whose attributes satisfy ar and then saves it as a boolean matrix X. Line 7 finds the frequent attribute set $FreAttrSet$ in X. Lines 8–9 get the candidate rules and the length and then sort them by number of attributes. Lines 10–21 screen candidate rules according to the confidence degree and threshold K and then obtain the rule set R.

4.5 Rule Pruning and Rule Optimization

The rule pruning and rule optimization algorithms are used to improve the quality of the ABAC policies. During the training, two or more sets may map to the same

rule. If there are two similar rules, the difficulty and complexity of policy mining are higher and may also affect the accuracy of the policy, which can reduce the quality of the policy. To solve this problem, we find similar rules, calculate their similarity, and then delete rules that do not affect the quality of the policy. If removing either of these rules does not improve the quality of the policy, we keep both rules. This may happen when there are two similar ABAC rules in the actual rule.

We use Jaccard similarity to measure the similarity between two rules, as follows:

Definition 16. Given two sets, A and B, the Jaccard coefficient is defined as the ratio of the size of the intersection of A and B and the size of the union of A and B. The calculation formula is as follows:

$$J(A,B) = \frac{|A \bigcap B|}{|A \bigcup B|} = \frac{|A \bigcap B|}{|A| + |B| - |A \bigcap B|}. \tag{19}$$

When sets A and B are empty, $J(A,B)$ is defined as 1.

According to the Theorem 16, we can calculate the similarity between rules r_1 and r_2. The formula is as follows:

$$J(r_1, r_2) = \frac{\sum_{v \in \{V(F), V(op)\}} |v_{r_1} \bigcap v_{r_2}|}{\sum_{v \in \{V(F), V(op)\}} |v_{r_1} \bigcup v_{r_2}|}, \tag{20}$$

where v represents their attribute domain, and the calculated results can determine their similarity. We consider that if the Jaccard similarity score is greater than 0.5, there is a significant overlap between them, and the two rules can be considered similar. This means that the size of their common elements is more than half the size of their union of elements. The Algorithm 3 gives the detailed procedure of rule pruning.

The details of the Algorithm 3 are as follows. The input is the set of rules R obtained by the Algorithm 2 and a similarity threshold δ, and the output is the trimmed rule R^*. Lines 3–21 calculate the similarity and select rules: If similarity of two rules is greater than or equal to a given threshold δ, put the two rules in set R_{t1}, and then, calculate the quality of $R - \{r_i\}$ to determine which can improve the quality of the policy, and save them to set R_{t2}. If there are multiple similar rules and their quality is greater than q, delete the one with the worst quality. Finally, we obtain a new rule set R^*.

Definition 17 (Rule Conflict Relation)**.** Given a rule set (R) and two rules $(r_1, r_2) \in R$, if $r_1 = r_2$, and one of them has the decision result of *permit* and the other has *deny*, then r_1 and r_2 have a conflicting relation.

Definition 18 (Rule Hierarchy Relation)**.** Given a rule set (R) and two rules (r_1, $r_2 \in R$), r_1 is called the senior rule of r_2, denoted as $r_1 \leq r_2$, and r_1 and r_2 have a rule hierarchical relation if $(F_{u_{r_1}} \subseteq F_{u_{r_2}}) \wedge (F_{o_{r_1}} \subseteq F_{o_{r_2}}) \wedge (F_{e_{r_1}} \subseteq F_{e_{r_2}})$, $op_{r_1} = op_{r_2}$,

Algorithm 3 Rule pruning algorithm

Input: R, δ // δ is a threshold, tentatively set at 0.5
Output: R^*

procedure PruneRule(R, δ)
$R^* \leftarrow \emptyset; R_{t1} \leftarrow \emptyset; R_{t2} \leftarrow \emptyset$
$q(R) \leftarrow$ *CalculateRuleQuality*(R) // $q(R)$ refers to the accuracy
for all $r_i \in R$ **do**
    for all $r_j \in R$ and $r_i \neq r_j$ **do**
        if Similarity$(r_i, r_j) \geq \delta$ **then**
            $R_{t1} \leftarrow R_{t1} \cup \{r_i, r_j\}$
        end if
    end for
end for
for all $r_i \in R_{t1}$ **do**
    $q(r_i) \leftarrow$ *CalculateRuleQuality*$(R - \{r_i\})$
    if $q(r_i) \geq q(R)$ **then**
        $R_{t2} \leftarrow R_{t2} \cup \{r_i\}$
    end if
end for
for all $r_i \in R_{t2}$ **do**
    if $q(r_i) == \min\{q(r_i), r_i \in R_{t2}\}$ **then**
        $R^* \leftarrow R - \{r_i\}$
    end if
end for
return R^*
end procedure

$d_{r_1} = d_{r_2}$. It includes rule-inclusion relation $((F_{u_{r_1}} \subset F_{u_{r_2}}) \wedge (F_{o_{r_1}} \subset F_{o_{r_2}}) \wedge (F_{e_{r_1}} \subset F_{e_{r_2}})$, $op_{r_1} = op_{r_2}$, $d_{r_1} = d_{r_2}$) and rule-equality relation $((F_{u_{r_1}} = F_{u_{r_2}}) \wedge (F_{o_{r_1}} = F_{o_{r_2}}) \wedge (F_{e_{r_1}} = F_{e_{r_2}})$, $op_{r_1} = op_{r_2}$, $d_{r_1} = d_{r_2}$).

If r_1 is senior to r_2, then r_2 is a more restrictive rule than r_1. If there are hierarchical rules r_1 and r_2 ($r_1 \leq r_2$) in the ABAC system, redundancy occurs, which leads to more FP in the decision. To reduce FP, we prune the extracted rule set by removing the overly permissive rule (r_1).

For example, here are two rules:

$$r_1 = (\{\langle Position, Student\rangle, \langle Location, Campus\rangle, \langle Type, Book\rangle\}, borrow, permit),$$

$$r_2 = (\{\langle Position, Student\rangle, \langle Location, Campus\rangle, \langle Time, 10{:}00\text{–}22{:}00\rangle, \langle Type, Book\rangle\}, borrow, permit).$$

Rule r_1 permits students to borrow books on campus, whereas rule r_2 permits students to borrow books on campus only during the specified time period. Here,

r_1 is the senior rule of r_2; so, r_2 is more restrictive. Therefore, access requests that would otherwise be denied are permitted owing to the overly permissive rule r_1, resulting in higher FP. To reduce the potential of more FP, we remove r_1 from the extracted rule set.

In the training process, owing to the lack of some samples in the training data or other reasons, some rules may be missing in the generated ABAC rules. Thus, some rules are ignored in the rule pruning process. This problem inevitably leads to FN because according to the missing rule, the access request that was originally permitted is denied owing to the generated rule. However, if some attribute relations are omitted in the process of extracting rules, for example, some attributes or attribute values of a rule are lost, the extracted rules are more relaxed than the actual rules, and the decisions that should be denied according to the actual rules are permitted instead. As mentioned in the above example, r_1 is more permissive than r_2, resulting in the FP phenomenon.

We provide rule optimization algorithms to solve the above problem, as shown in Algorithm 4. This process is similar to the training process in ML, where the training data includes the access control decisions that produce FP or FN. For example, in the FP scenario, a request should be denied according to the actual policy. Despite this, it is permitted according to the generated policy owing to the extraction of overly permissive rules in the policy generation process. In the FN scenario, a request should be permitted according to the actual policy. Moreover, it is denied according to the generated policy, which is closely related to the missing samples in the data.

The Algorithm 4 is described as follows. Through k-fold cross-validation, we obtain classification evaluation indicators (such as *FNR*, *FPR*, and *Acc*) to determine the generated rule results. In addition, we check if there are hierarchical (inclusive or equal) relations and conflicting relations among the rules using the accuracy. Lines 7–25 handle rules with hierarchy and conflict by preserving one of the equal rules and deleting the senior rule and the conflict rules. We finally achieve an optimized rule set R_{Q_p}.

For policies generated on the access logs, to improve the quality of mining the ABAC policy, we combined *WSC* and *Accuracy* (*Acc*) to define the comprehensive indicator of policy quality. The formula is as follows:

$$Q_p = \frac{1}{\frac{\alpha}{Acc} + \frac{1-\alpha}{\Delta WSC}}, \tag{21}$$

where ΔWSC is calculated as follows:

$$\Delta WSC = \frac{WSC_{max} - WSC_{\pi}}{WSC_{max} - WSC_{min}}. \tag{22}$$

When $WSC_{\pi} = WSC_{max}$, $\Delta WSC = 0$, define $Q_p = 0$. When $WSC_{\pi} = WSC_{min}$, $\Delta WSC = 1$, it indicates no policy generated; so, also define $Q_p = 0$. Let $\alpha = \frac{1}{1+\beta^2}$, $\beta \in \mathbb{R}$ in (21), then β determines the importance degree of *Acc* to

Algorithm 4 Rule optimization algorithm

Input: D, L, R^*
Output: R_{Q_p}

1: **procedure** OptimizeRule(D, L, R^*)
2: $Acc \leftarrow GetIndicator_C(D, L, R^*)$
3: $WSC \leftarrow GetWSC(R^*)$
4: $\Delta WSC \leftarrow GetIndicator_W(R^*, WSC)$
5: $Q_p \leftarrow FindParameter(Acc, \Delta WSC)$
6: $R_{Q_p} \leftarrow FilterRule(R^*, Q_p)$
7: **for** all $r_o \in R_{Q_p}$ **do**
8: $R_o \leftarrow FindRule(R_{Q_p}, r_o)$
9: **if** $R_o \neq \emptyset$ **then**
10: **for** all $r_i, r_j \in R_o$ **do**
11: **if** $r_i == r_j$ **then**
12: $R_{Q_p} \leftarrow R_{Q_p} - (\{r_i\}\ or\ \{r_j\})$
13: **end if**
14: **if** $r_j \leq r_i$ **then**
15: $R_{Q_p} \leftarrow R_{Q_p} - \{r_j\}$
16: **else**
17: $R_{Q_p} \leftarrow R_{Q_p} - \{r_i\}$
18: **end if**
19: **if** r_i Conflicts with r_j **then**
20: $R_{Q_p} \leftarrow R_{Q_p} - \{r_i\} - \{r_j\}$
21: **end if**
22: **end for**
23: **end if**
24: **end for**
25: **return** R_{Q_p}
26: **end procedure**

the policy complexity. When $\beta = 1$, the two indicators Acc and ΔWSC have the same weight, indicating the same importance. When $\beta < 1$, the weight of Acc is significant, indicating that Acc is more important. When $\beta > 1$, ΔWSC has a significant weight, meaning that ΔWSC is more important. ΔWSC is the normalized value of the WSC, putting the ΔWSC value in the interval [0, 1]. WSC_{max} and WSC_{min} represent the complexity of the weighted structure of the most complex and simplest policies, respectively. The most complex policy can be understood as the policy corresponding to each access control decision that contains all attributes of the subject, object, and environment. The simplest policy can be understood as the null policy, namely $WSC_{min} = 0$.

In addition, the Algorithm 5 shows the policy generation step in detail. The Algorithm 5 is described as follows. Line 2 is for preprocessing the access logs. Line 3 is for selecting attributes. Lines 4–5 are for extracting and pruning rules.

Line 6 calculates the policy/rule quality indicator $Q(WSC, Acc)$. Finally, lines 7–10 are used to refine the mined rules until the best rule set is obtained.

Algorithm 5 CatBoost-based ABAC policy generation algorithm

Input: L, m, D, T, δ, K, Q // Q refers to the policy/rule quality indicator (WSC, Acc)

Output: R_e

1: **procedure** GenerateABACRule$(L, m, D, T, \delta, K, Q)$
2: $A \leftarrow Preprocess(L)$
3: $A^* \leftarrow SelectAttribute(L, A, m)$
4: $R \leftarrow ExtractRule(A^*, L, D, T, K)$
5: $R^* \leftarrow PruneRule(R, \delta)$
6: $q \leftarrow CalculateRuleQuality(R^*)$
7: **while** $q < Q$ or $Acc \geq 0.95$ **do**
8: $\quad R_{Q_p} \leftarrow OptimizeRule(D, L, R^*)$
9: $\quad q \leftarrow CalculateRuleQuality(R_{Q_p})$
10: **end while**
11: **return** R_{Q_p}
12: **end procedure**

Some parameters are additionally used to adjust the model to improve the accuracy. We use $p(0)$ and $p(1)$ to denote the probability estimations of the *permit* and *deny* decision against the access requests, respectively. If $p(1) > p(0)$, the decision is *permit*; if $p(1) < p(0)$, the decision is *deny*; and if $p(1) = p(0)$, the decision cannot be determined.

Next, we use the cross entropy to calculate the loss of this model because the training goal is to minimize the cross entropy of the two categories (*permit* and *deny*). The calculation formula is as follows:

$$LF = -\sum_{i=1}^{n}\{w(0) \cdot y_i(0) \cdot \log[p_i(0)] + w(1) \cdot y_i(1) \cdot \log[p_i(1)]\}, \tag{23}$$

where n is the number of access requests in the training process; and $y_i(0)$, $y_i(1)$ represent the decision to *deny* and *permit*, respectively.

If the decision results are wrong, $\lceil y_i(0), y_i(1) \rfloor = \lceil 1, 0 \rfloor$. If decision results are correct, $\lceil y_i(0), y_i(1) \rfloor = \lceil 0, 1 \rfloor$. When the probability estimations of $y_i(0)$ and $y_i(1)$ are $p_i(0)$ and $p_i(1)$, respectively, and any decision result is correct, for all access requests, each decision completely matches the actual decision. This indicates that the loss function has reached its minimum absolute value and is equal to 0. In the loss function, to balance the *permit* and *deny* decision results in the data training process, we define the weights $w(1) = \frac{(N-N_1)}{N}$ and $w(0) = 1 - w(1)$, where $w(1)$ and $w(0)$ are the weights of *permit* and *deny*, respectively. N is the input quantity of the dataset, and N_1 is the number of *permit* decisions.

5 EVALUATION

5.1 Simple Evaluation

We compared the ABAC policy generation approach from access logs based on the CatBoost with five related approaches in the following nine aspects. The comparison results are shown in Table 3.

	Xu et al. [23]	Medvet et al. [28]	Iyer et al. [24]	Cotrini et al. [30]	Mocanu et al. [38]	Ours
Attribute relation	no	no	no	no	no	yes
Negative decision rule	no	no	yes	no	no	yes
Sparse logs	no	yes	no	yes	yes	yes
Noise logs	yes	no	no	no	yes	yes
WSC	yes	yes	yes	no	yes	yes
Policy accuracy	yes	yes	yes	yes	no	yes
Policy complexity	yes	yes	yes	yes	yes	yes
Attribute and operation weight	no	no	no	no	no	yes
Policy quality comprehensive indicator	no	no	no	no	no	yes

Table 3. Comparison of ABAC policy mining approaches

5.2 Experimental Evaluation

5.2.1 Dataset Introduction and Experimental Settings

The experimental environment was set as follows. The processor was based on X64, and the parameters were Intel(R) Core(TM) I7-8565U CPU @ 1.80 GHz 1.99 GHz. The random access memory (RAM) was 8 GB. The operating system was Windows 10 Home version (64-bit). The version number was 21H1. The experimental platform was Anaconda 3-2022.05 version, and the interpreter was Python version 3.9.12. The ABAC policy generation approach was implemented based on CatBoost.

The experimental dataset was from the "Amazon.com-Employee Access Challenge" competition on the Kaggle platform, divided into the training set and test set. For short, this is called the Amazon-employee dataset.[2] This consists of real historical data from 2010 and 2011. Each access tuple in this dataset comprises the tuple corresponding to an employee's access request to a resource and displays the corresponding decision (*permit* or *deny*) result. Its access log is composed of employee attribute values and resource identifiers. It has many subject attributes but a relatively small number of log entries and a sparse log set. Moreover, there is

[2] `https://www.kaggle.com/c/amazon-employee-access-challenge/forums/t/5283/winning-solution-code-and-methodology`

only one object attribute, which may lead to biased experimental results. Table 4 shows the basic information of the training set in this dataset; there are a total of 32 769 access control records.

Attribute Name	Attribute Type	Attribute Information	Attribute Number
ACTION	Operation attribute	ACTION is 1 if the resource was approved and 0 if not.	2
RESOURCE	Object attribute	An ID for each resource	7 518
MGR_ID	Subject attribute	The EMPLOYEE ID of the manager of the current EMPLOYEE ID record; an employee may have only one manager at a time	4 243
ROLE_ROLLUP_1	Subject attribute	Company role grouping category id1 (e.g. US Engineering)	128
ROLE_ROLLUP_2	Subject attribute	Company role grouping category id2 (e.g. US Retail)	177
ROLE_DEPTNAME	Subject attribute	Company role department description (e.g. Retail)	449
ROLE_TITLE	Subject attribute	Company role business title description (e.g. e.g. Senior Engineering, Retail Manager)	343
ROLE_FAMILY_DESC	Subject attribute	Company role family extended description (e.g. Retail Manager, Software Engineering)	2 358
ROLE_FAMILY	Subject attribute	Company role family description (e.g. Retail Manager)	67
ROLE_CODE	Subject attribute	Company role code: this code is unique to each role (e.g. Manager)	343

Table 4. Dataset information

5.2.2 Experiments and Comparison

Through experiments, we compared our approach with [38], which used the RBM algorithm to infer ABAC policies from logs, and [39], which used the MLP algorithm to infer the ABAC policies and made a detailed comparison in the following aspects. Finally, our approach is superior to theirs in most cases, but it needs to be revised in some cases, and it will be analyzed in detail.

Figure 3 compares the attribute importance of the two approaches. It can be seen that after using different approaches, there are apparent differences in the importance of different attributes. For example, it can be seen that the importance of RESOURCE attribute in the [38], which used RBM is almost 0. In our approach,

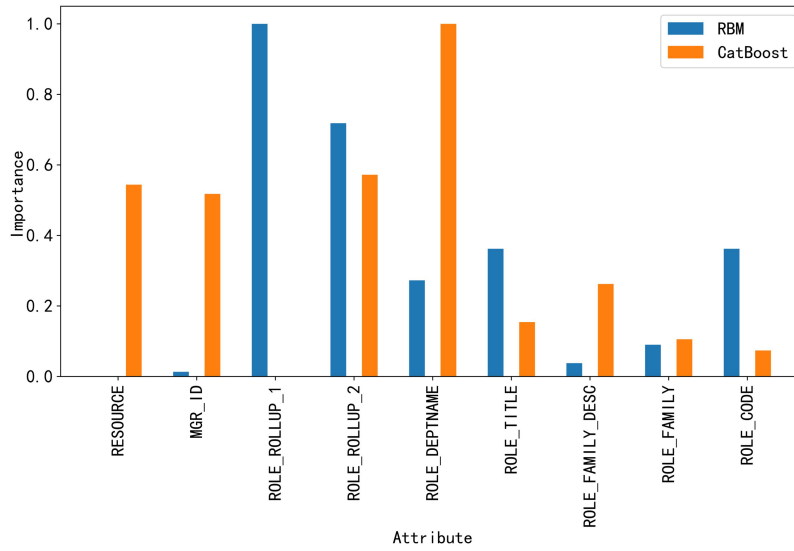


Figure 3. Attribute importance comparison graph

it is 0.77. The attribute ROLE_DEPTNAME has the highest importance, close to 1. The attribute ROLE_ROLLUP_1 has the lowest importance, which is almost 0. The attribute RESOURCE belongs to the object attribute, which is single and extremely important. The attribute ROLE_ROLLUP_1 belongs to the subject attributes and has the lowest importance. Because the MLP algorithm used in [39] is a nonlinear classifier, it cannot analyze the importance of attributes, so it cannot weigh them. But, of course, it can be understood that the weights are the same, and all are equal (specified as 1).

Through the k-fold cross-validation method, we divided the training set in the Amazon Employee dataset into five parts on average, using 1 part as the training set each time and the remaining four parts as the test set. After five times averaging, we obtained our final experimental results. Figure 4 shows the receiver operating characteristic (ROC) curves of the three approaches, which show the accuracy of each approach's access control decision results. According to the area enclosed by the ROC curve and the coordinate axis, that is, the area under curve (AUC) value, we can see that the AUC value obtained by our approach is 0.978. It is slightly higher than the AUC value 0.972, obtained by the RBM algorithm in [38], and the AUC value 0.97, obtained by the MLP algorithm in [39], respectively. It indicates that the accuracy of the decision results obtained by our approach is higher than that obtained by the [38] and [39] approaches.

Figure 5 shows the relations between the access control decision precision and the recall rate. It indicates that the decision precision decreases with the increase of recall rate. It can be seen that there is a turning point when the recall rate is

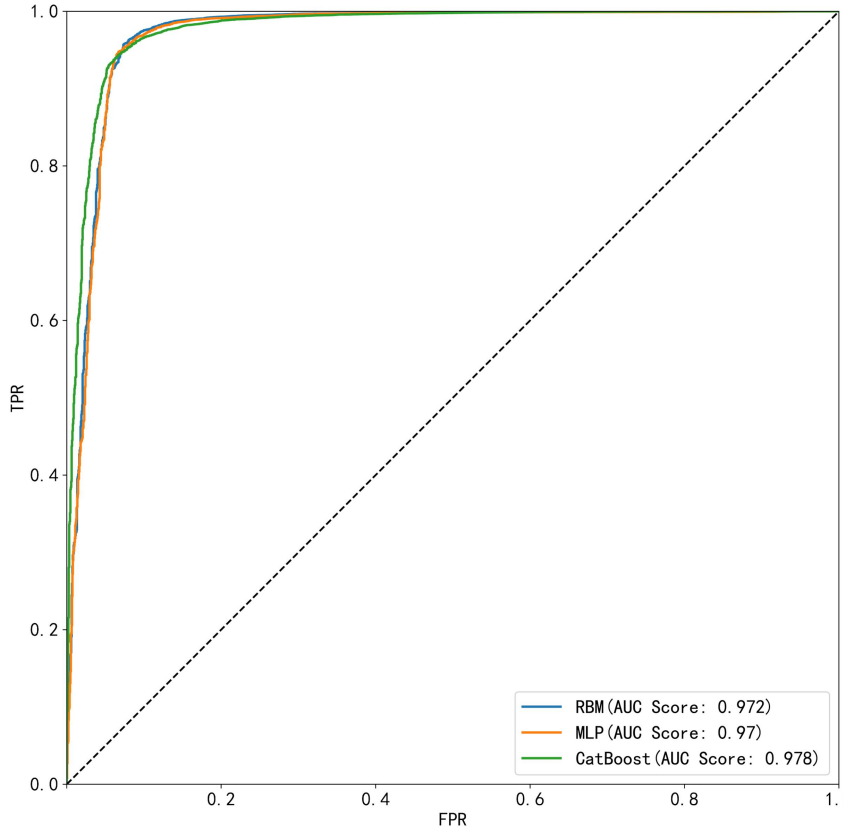


Figure 4. ROC curve

about 0.95. When the recall rate is less than 0.95, the decision precision of our approach is higher than that obtained by using the RBM algorithm and the MLP algorithm in [39]. When the recall rate is greater than 0.95, the decision precision of our approach is slightly lower than that of the approaches in [38] and [39]. On the whole, the decision precision of our approach is better than theirs. By comparing the precision and the value of AUC, we can see that the decision result of our approach is superior to that of [38] and [39].

Our approach is compared with the decision results obtained using the RBM algorithm in [38] and the MLP algorithm in [39] through k-fold cross-validation. In Table 5, it was evident that our approach is superior to the approach used in [38] and [39] in terms of *TPR* and *FNR* in both the test set and training set and inferior to the approach used in [38] and [39] in terms of *TNR*, *FPR*, and *Mcc*. In terms of *precision*, our approach is slightly inferior to that in [38] and [39]. Only in the test set, our approach is slightly superior to that in [38]. In the training set, the

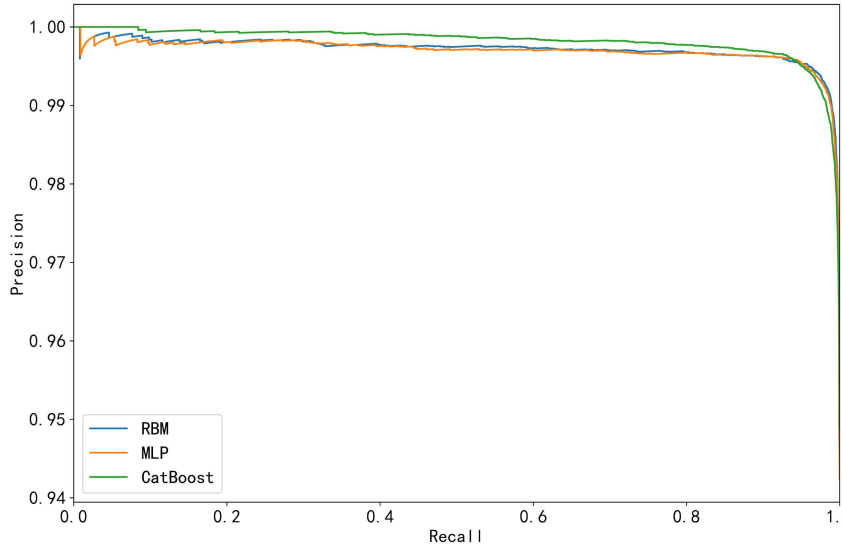


Figure 5. P-R curve

values of *Acc*, *F1-score*, and the AUC of our approach are slightly inferior to those used in [38] and [39]. However, in the test set, our approach is superior to theirs. The decision accuracy of our approach in the test set is 95.74 %. It is higher than 94.54 % in [38] and 93.13 % in [39]; the AUC value in the test set is 90.37 %, which is much higher than 84.87 % in [38] and 84.02 % in [39].

Approach	TPR	TNR	FPR	FNR
RBM-training	0.996312	0.903896	0.096104	0.003688
RBM-test	0.978699	0.366947	0.633053	0.021301
MLP-training	0.992705	0.945455	0.054545	0.007295
MLP-test	0.959819	0.436975	0.563025	0.040181
CatBoost-training	0.997933	0.672078	0.327922	0.002067
CatBoost-test	0.991770	0.361345	0.638655	0.008230

Approach	Precision	Accuracy	F1-score	Mcc	AUC
RBM-training	0.994016	0.990883	0.995163	0.916287	0.998517
RBM-test	0.964076	0.945377	0.971332	0.399686	0.848675
MLP-training	0.996582	0.989929	0.994640	0.911994	0.998199
MLP-test	0.967312	0.931340	0.963551	0.373990	0.840196
CatBoost-training	0.979904	0.978791	0.988836	0.790621	0.994601
CatBoost-test	0.964230	0.957431	0.977806	0.490341	0.903660

Table 5. Comparison of the predicted results for access control decisions

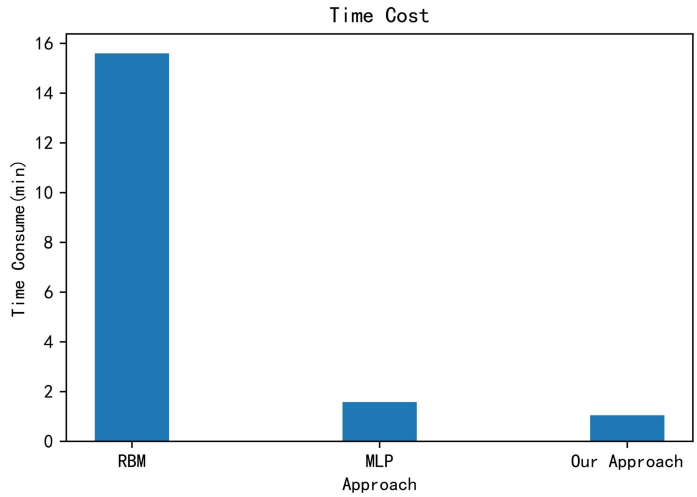


Figure 6. Comparison of time consumption

Figure 6 shows the comparison of the time consumed by our approach and the two approaches used in [38] and [39] in making decisions on test sets. Through cross-validation, we made access control decision predictions on the training set containing 80 %, with 26 215 access control records. Our approach consumes less time than the approaches used in the [38] and the [39].

5.2.3 Result Analysis

By comparing various aspects of the three approaches, it is reasonable to choose the ABAC policy generation approach based on the CatBoost. Figure 7 shows the loss function obtained by our approach in the training set. It can be seen that with the increase in the number of iterations, the value of the loss function gradually decreases and tends to be stable when the number of iterations is about 3 000. Therefore, we decide to use the number of iterations to conduct the final experiment 3 000 times.

Figures 8 and 9 represent the confusion matrix of the access logs before and after rule pruning and rule optimization, respectively. As shown in Figures 8 and 9, the access control decision prediction was made on the training set containing 80 %, and there were a total of 26 215 access control records. After pruning and optimization, the number of TP records increased from 24 597 to 24 624; the number of TN records increased from 900 to 1 035; and the accuracy improved considerably. The number of wrong decisions decreased by 162; FP records decreased from 640 to 505; and FN records decreased from 78 to 51. The accuracy rate of the decision results was improved from 97.46 % to 98.00 %.

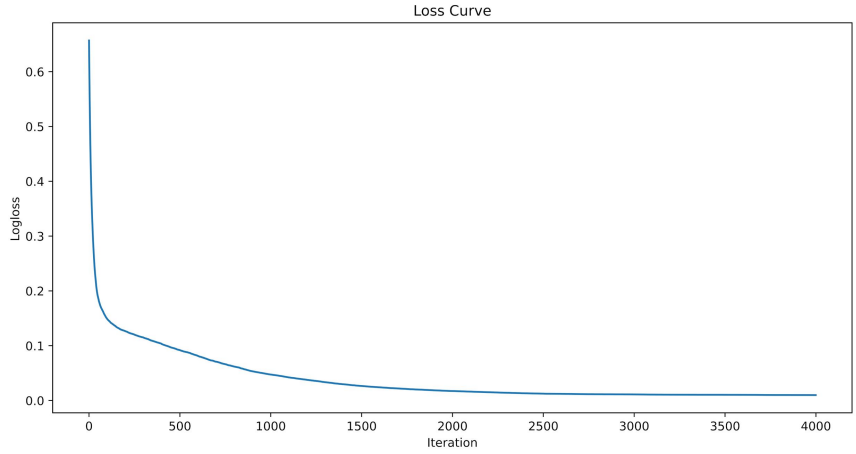


Figure 7. Loss iteration graph

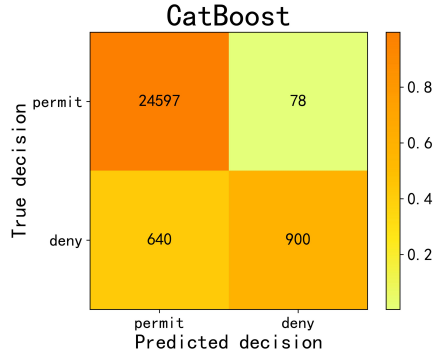


Figure 8. Before rule pruning and optimization

To sum up, the ABAC policy generation approach based on the CatBoost algorithm is a decision approach to predict access decisions according to historical access logs (or access control records). Through experiments, we compared it with the two approaches used in [38] and [39]. The *precision* and Mcc are inferior to these approaches, but our approach is superior to them in other aspects, especially the accuracy of the decision results and the time consumption. However, our approach assumes that the decision result is only *permit* or *deny*. In the actual access control scenario, there will be more complex decision results, such as decision conflict (that is, both *permit* and *deny*), which is the disadvantage of this approach.

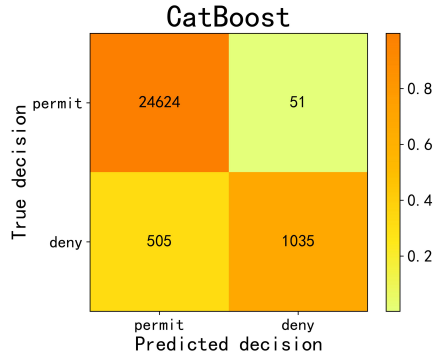


Figure 9. After rule pruning and optimization

6 CONCLUSION

This paper provided an approach to automate ABAC policy (rule) generation via the CatBoost ML algorithm. This approach can discover both positive and negative ACPs. We presented an attribute selection algorithm by weighted reconstruction of the attributes in the quasi-generation policy, thus improving the validity of rule extraction. The rule extraction algorithm, rule pruning algorithm, and rule optimization algorithm were also proposed to improve the precision of the generated policy and significantly improve the accuracy of the generated policy. Most importantly, we proposed a new policy quality indicator, namely the policy quality comprehensive indicator, to measure the accuracy and simplicity of the policy. It is essential to compare the generated policy with the actual policy for further refinement. We evaluated the presented approach on the Amazon-employee dataset and verified its feasibility, effectiveness, and practicability. Finally, through experiments, we demonstrated that although *FPR*, *precision*, and other aspects are slightly inferior to approaches [38] and [39], our approach is superior to theirs in terms of the accuracy and time consume of the generated policy.

In future work, we will continue to improve our approach's accuracy and simplicity, further study how to resolve conflicts and undecidability in access control decisions and research other factors that influence the quality of generated policies.

Acknowledgements

This research was supported by the National Natural Science Foundation of China (Grant No. 61862059).

Conflicts of interest

We declare that we have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

REFERENCES

[1] Feng, D.—Zhang, M.—Li, H.: Big Data Security and Privacy Protection. Chinese Journal of Computers, Vol. 37, 2014, No. 1, pp. 246–258, doi: 10.3724/SP.J.1016.2014.00246 (In Chinese).

[2] Li, H.—Zhang, M.—Feng, D.—Hui, Z.: Research on Big Data Access Control. Chinese Journal of Computers, Vol. 40, 2017, No. 1, pp. 72–91, doi: 10.11897/SP.J.1016.2017.00072 (In Chinese).

[3] Graham, G. S.—Denning, P. J.: Protection: Principles and Practice. Proceedings of the May 16-18, 1972, Spring Joint Computer Conference (AFIPS '72 (Spring)), ACM, 1971, pp. 417–429, doi: 10.1145/1478873.1478928.

[4] Sandhu, R. S.—Samarati, P.: Access Control: Principles and Practice. IEEE Communications Magazine, Vol. 32, 1994, No. 9, pp. 40–48, doi: 10.1109/35.312842.

[5] Sandhu, R. S.: Lattice-Based Access Control Models. Computer, Vol. 26, 1993, No. 11, pp. 9–19, doi: 10.1109/2.241422.

[6] Sandhu, R. S.—Coyne, E. J.—Feinstein, H. L.—Youman, C. E.: Role-Based Access Control Models. Computer, Vol. 29, 1996, No. 2, pp. 38–47, doi: 10.1109/2.485845.

[7] Hu, V. C.—Ferraiolo, D.—Kuhn, R.—Schnitzer, A.—Sandlin, K.—Miller, R.—Scarfone, K.: Guide to Attribute Based Access Control (ABAC) Definition and Considerations. NIST Special Publication 800-162. National Institute of Standards and Technology, Gaithersburg, MD, 2014, doi: 10.6028/NIST.SP.800-162.

[8] Wang, X.—Fu, H.—Zhang, L.: Research Progress on Attribute-Based Access Control. Acta Electronica Sinica, Vol. 38, 2010, No. 7, pp. 1660–1667 (In Chinese).

[9] Fang, L.—Yin, L.—Guo, Y.—Fang, B.: A Survey of Key Technologies in Attribute-Based Access Control Scheme. Chinese Journal of Computers, Vol. 40, 2017, No. 7, pp. 1680–1698, doi: 10.11897/SP.J.1016.2017.01680 (In Chinese).

[10] Das, S.—Mitra, B.—Atluri, V.—Vaidya, J.—Sural, S.: Policy Engineering in RBAC and ABAC. In: Samarati, P., Ray, I., Ray, I. (Eds.): From Database to Cyber Security: Essays Dedicated to Sushil Jajodia on the Occasion of His 70th Birthday. Springer, Cham, Lecture Notes in Computer Science, Vol. 11170, 2018, pp. 24–54, doi: 10.1007/978-3-030-04834-1_2.

[11] Xu, Z.—Stoller, S. D.: Mining Attribute-Based Access Control Policies from Logs. In: Atluri, V., Pernul, G. (Eds.): Data and Applications Security and Privacy XXVIII (DBSec 2014). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 8566, 2014, pp. 276–291, doi: 10.1007/978-3-662-43936-4_18.

[12] Vaidya, J.—Atluri, V.—Guo, Q.: The Role Mining Problem: Finding a Minimal Descriptive Set of Roles. Proceedings of the 12th ACM Symposium on Access Control Models and Technologies (SACMAT '07), 2007, pp. 175–184, doi: 10.1145/1266840.1266870.

[13] Molloy, I.—Chen, H.—Li, T.—Wang, Q.—Li, N.—Bertino, E.—Calo, S.—Lobo, J.: Mining Roles with Multiple Objectives. ACM Transactions on Information and System Security, Vol. 13, 2010, No. 4, Art. No. 36, doi: 10.1145/1880022.1880030.

[14] Molloy, I.—Li, N.—Qi, Y.—Lobo, J.—Dickens, L.: Mining Roles with Noisy Data. Proceedings of the 15th ACM Symposium on Access Control Models and Technologies (SACMAT '10), 2010, pp. 45–54, doi: 10.1145/1809842.1809852.

[15] Currey, J.—McKinstry, R.—Dadgar, A.—Gritter, M.: Informed Privilege-Complexity Trade-Offs in RBAC Configuration. Proceedings of the 25th ACM Symposium on Access Control Models and Technologies (SACMAT '20), 2020, pp. 119–130, doi: 10.1145/3381991.3395597.

[16] Jafarian, J. H.—Takabi, H.—Touati, H.—Hesamifard, E.—Shehab, M.: Towards a General Framework for Optimal Role Mining: A Constraint Satisfaction Approach. Proceedings of the 20th ACM Symposium on Access Control Models and Technologies (SACMAT '15), 2015, pp. 211–220, doi: 10.1145/2752952.2752975.

[17] Molloy, I.—Park, Y.—Chari, S.: Generative Models for Access Control Policies: Applications to Role Mining over Logs with Attribution. Proceedings of the 17th ACM Symposium on Access Control Models and Technologies (SACMAT '12), 2012, pp. 45–56, doi: 10.1145/2295136.2295145.

[18] Narouei, M.—Takabi, H.: Towards an Automatic Top-Down Role Engineering Approach Using Natural Language Processing Techniques. Proceedings of the 20th ACM Symposium on Access Control Models and Technologies (SACMAT '15), 2015, pp. 157–160, doi: 10.1145/2752952.2752958.

[19] Narouei, M.—Takabi, H.: Automatic Top-Down Role Engineering Framework Using Natural Language Processing Techniques. In: Akram, R. N., Jajodia, S. (Eds.): Information Security Theory and Practice (WISTP 2015). Springer, Cham, Lecture Notes in Computer Science, Vol. 9311, 2015, pp. 137–152, doi: 10.1007/978-3-319-24018-3_9.

[20] Anderer, S.—Scheuermann, B.—Mostaghim, S.—Bauerle, P.—Beil, M.: RMPlib: A Library of Benchmarks for the Role Mining Problem. Proceedings of the 26th ACM Symposium on Access Control Models and Technologies (SACMAT '21), 2021, pp. 3–13, doi: 10.1145/3450569.3463566.

[21] Chari, S. N.—Molloy, I. M.: Generation of Attribute Based Access Control Policy from Existing Authorization System. Google Patents, 2016 (US Patent US9264451B2).

[22] Xu, Z.—Stoller, S. D.: Mining Attribute-Based Access Control Policies from RBAC Policies. 2013 10th International Conference and Expo on Emerging Technologies for a Smarter World (CEWIT), IEEE, 2013, pp. 1–6, doi: 10.1109/CEWIT.2013.6713753.

[23] Xu, Z.—Stoller, S. D.: Mining Attribute-Based Access Control Policies. IEEE

Transactions on Dependable and Secure Computing, Vol. 12, 2015, No. 5, pp. 533–545, doi: 10.1109/TDSC.2014.2369048.

[24] Iyer, P.—Masoumzadeh, A.: Mining Positive and Negative Attribute-Based Access Control Policy Rules. Proceedings of the 23rd ACM Symposium on Access Control Models and Technologies (SACMAT '18), 2018, pp. 161–172, doi: 10.1145/3205977.3205988.

[25] Chakraborty, S.—Sandhu, R.—Krishnan, R.: On the Feasibility of Attribute-Based Access Control Policy Mining. 2019 20th IEEE International Conference on Information Reuse and Integration for Data Science (IRI), 2019, pp. 245–252, doi: 10.1109/IRI.2019.00047.

[26] Talukdar, T.—Batra, G.—Vaidya, J.—Atluri, V.—Sural, S.: Efficient Bottom-Up Mining of Attribute Based Access Control Policies. 2017 IEEE 3rd International Conference on Collaboration and Internet Computing (CIC), 2017, pp. 339–348, doi: 10.1109/CIC.2017.00051.

[27] Narouei, M.—Takabi, H.: A Nature-Inspired Framework for Optimal Mining of Attribute-Based Access Control Policies. Security and Privacy in Communication Networks (SecureComm 2019), Springer, Cham, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Vol. 305, 2019, pp. 489–506, doi: 10.1007/978-3-030-37231-6_29.

[28] Medvet, E.—Bartoli, A.—Carminati, B.—Ferrari, E.: Evolutionary Inference of Attribute-Based Access Control Policies. In: Gaspar-Cunha, A., Henggeler Antunes, C., Coello, C. C. (Eds.): Evolutionary Multi-Criterion Optimization (EMO 2015). Springer, Cham, Lecture Notes in Computer Science, Vol. 9018, 2015, pp. 351–365, doi: 10.1007/978-3-319-15934-8_24.

[29] Das, S.—Sural, S.—Vaidya, J.—Atluri, V.: Using Gini Impurity to Mine Attribute-Based Access Control Policie with Environment Attributes. Proceedings of the 23rd ACM Symposium on Access Control Models and Technologies (SACMAT '18), 2018, pp. 213–215, doi: 10.1145/3205977.3208949.

[30] Cotrini, C.—Weghorn, T.—Basin, D.: Mining ABAC Rules from Sparse Logs. 2018 IEEE European Symposium on Security and Privacy, 2018, pp. 31–46, doi: 10.1109/EuroSP.2018.00011.

[31] Karimi, L.—Joshi, J.: An Unsupervised Learning Based Approach for Mining Attribute Based Access Control Policies. 2018 IEEE International Conference on Big Data (Big Data), 2018, pp. 1427–1436, doi: 10.1109/BigData.2018.8622037.

[32] Das, S.—Sural, S.—Vaidya, J.—Atluri, V.—Rigoll, G.: VisMAP: Visual Mining of Attribute-Based Access Control Policies. In: Garg, D., Kumar, N. V. N., Shyamasundar, R. K. (Eds.): Information Systems Security (ICISS 2019). Springer, Cham, Lecture Notes in Computer Science, Vol. 11952, 2019, pp. 79–98, doi: 10.1007/978-3-030-36945-3_5.

[33] Narouei, M.—Khanpour, H.—Takabi, H.—Parde, N.—Nielsen, R.: Towards a Top-Down Policy Engineering Framework for Attribute-Based Access Control. Proceedings of the 22nd ACM Symposium on Access Control Models and Technologies (SACMAT '17), 2017, pp. 103–114, doi: 10.1145/3078861.3078874.

[34] Narouei, M.—Takabi, H.—Nielsen, R.: Automatic Extraction of Access

Control Policies from Natural Language Documents. IEEE Transactions on Dependable and Secure Computing, Vol. 17, 2020, No. 3, pp. 506–517, doi: 10.1109/TDSC.2018.2818708.

[35] Alohaly, M.—Takabi, H.—Blanco, E.: A Deep Learning Approach for Extracting Attributes of ABAC Policies. Proceedings of the 23rd ACM Symposium on Access Control Models and Technologies (SACMAT '18), 2018, pp. 137–148, doi: 10.1145/3205977.3205984.

[36] Alohaly, M.—Takabi, H.—Blanco, E.: Automated Extraction of Attributes from Natural Language Attribute-Based Access Control (ABAC) Policies. Cybersecurity, Vol. 2, 2019, Art. No. 2, doi: 10.1186/s42400-018-0019-2.

[37] Karimi, L.—Abdelhakim, M.—Joshi, J.: Adaptive ABAC Policy Learning: A Reinforcement Learning Approach. CoRR, 2021, doi: 10.48550/arXiv.2105.08587.

[38] Mocanu, D. C.—Turkmen, F.—Liotta, A.: Towards ABAC Policy Mining from Logs with Deep Learning. Proceedings of the 18th International Multiconference - Intelligent Systems (IS 2015), 2015.

[39] Cappelletti, L.—Valtolina, S.—Valentini, G.—Mesiti, M.—Bertino, E.: On the Quality of Classification Models for Inferring ABAC Policies from Access Logs. 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 4000–4007, doi: 10.1109/BigData47090.2019.9005959.

[40] Prokhorenkova, L.—Gusev, G.—Vorobev, A.—Dorogush, A. V.—Gulin, A.: CatBoost: Unbiased Boosting with Categorical Features. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (Eds.): Advances in Neural Information Processing Systems 31 (NeurIPS 2018). Curran Associates, Inc., 2018, pp. 6639–6649.

[41] Dorogush, A. V.—Ershov, V.—Gulin, A.: CatBoost: Gradient Boosting with Categorical Features Support. Corr, 2018, doi: 10.48550/arXiv.1810.11363.

[42] Hancock, J. T.—Khoshgoftaar, T. M.: CatBoost for Big Data: An Interdisciplinary Review. Journal of Big Data, Vol. 7, 2020, Art. No. 94, doi: 10.1186/s40537-020-00369-8.

[43] Tan, P.—Steinbach, M. S.—Karpatne, A.—Kumar, V.: Introduction to Data Mining (second Edition). Pearson, 2019, `https://www-users.cse.umn.edu/%7Ekumar001/dmbook/index.php`.

Shan Quan is currently a graduate student in the College of Mathematics and System Science, Xinjiang University, China. His research mainly focuses on statistics and information security.

Yongdan Zhao is currently a graduate student in the College of Mathematics and System Science, Xinjiang University, China. Her research mainly focuses on statistics and information security.

Nurmamat Helil received his B.Sc., M.Sc. and Ph.D. degrees in the School of Mathematical Sciences, Peking University in 2000, 2003 and 2008, respectively. He is Full Professor of the College of Mathematics and System Science, Xinjiang University, China. From April 2010 to April 2011, he worked as a post-doctor in the School of Computer Science and Engineering, Chung-Ang University, Korea. From April 2016 to April 2017, he worked as Visiting Research Scholar in the Department of Computer Science and Engineering, University of Minnesota Twin Cities, USA. His research interests include information system security, access control, and cloud storage security.

Computing and Informatics, Vol. 42, 2023, 651–666, doi: 10.31577/cai_2023_3_651

CLASSIFICATION OF SENTIMENT USING OPTIMIZED HYBRID DEEP LEARNING MODEL

Chaima Ahle Touate, Rachid El Ayachi, Mohamed Biniz

Faculty of Science and Technology
Sultan Moulay Slimane University
Mghila P3225
230 00 Beni Mellal, Morocco
e-mail: ahle.touate.chaima@gmail.com, rachid.elayachi@usms.ma, mohamedbiniz@gmail.com

Abstract. Sentiment classification plays a pivotal role in natural language processing (NLP), and prior research has established the efficacy of utilizing convolutional neural networks (CNNs) and long short-term memory (LSTM) in this task. However, these approaches suffer from individual performance limitations: CNNs are limited to extracting local information and fail to express context information adequately, while LSTM networks excel at extracting context dependencies but exhibit long training times. To address this issue, we propose a novel text classification algorithm based on a hybrid CNN-LSTM model that leverages the strengths of both approaches and overcomes their limitations by combining them. Our approach is evaluated on the IMDB dataset, and we present a hyperparameter optimization framework utilizing Random Search to increase the likelihood of producing an optimally performing model.

Keywords: Document classification, CNN, LSTM, hybrid models, hyperparameter tuning, random search

1 INTRODUCTION

Text classification has garnered significant attention in recent years and is considered one of the fundamental tasks in natural language processing (NLP) with various applications, such as sentiment analysis and topic labeling.

Traditional text classification methods rely on statistics and feature selection [1], which include commonly used algorithms like Naive Bayes [2], Support Vector Machine (SVM) [3], Decision Trees [4], and others. These classic machine-learning techniques have achieved remarkable results in text classification tasks. However, the introduction of text convolutional neural networks by Yoon Kim has led to the emergence of a variety of deep learning text classification methods [5]. This demonstrates the feasibility of applying artificial neural networks, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), to the field of text classification.

Although CNNs have the potential to extract local information, they may fail to capture long-distance dependencies. On the other hand, LSTMs can address this limitation by modeling texts sequentially across sentences. Despite the development of NN-based and word-embedding techniques, sentiment analysis remains challenging [5].

In this study, we propose a hybrid CNN-LSTM model consisting of two parts, CNN and LSTM, to predict the sentiment expressed in texts. We add Hybrid Attention to fully exploit the respective advantages of CNNs and LSTMs and fill their gaps by selectively learning long sequences and making deep neural networks in each training batch. Our proposed model can learn distinct feature forms, improve model learning and expression skills, and prevent overfitting.

The remainder of this paper is structured as follows: Section 2 reviews related works, Section 3 details the architecture of the developed classification system, Section 4 outlines the Hyper-parameters tuning, Section 5 presents the experimental results, Section 6 exhibits the evaluation methods followed by Section 7, which discuss the results, and finally, a conclusion in Section 8.

2 RELATED WORKS

The field of text classification has seen a surge in interest due to the advent of deep learning techniques that require less feature engineering and have the potential to achieve high accuracy. Yoon Kim introduced the convolutional neural network (CNN) to text classification and showed its ability to capture local correlations in the sentence through multiple kernels of varying sizes [6]. Since then, many researchers have proposed CNN-based models, but it was found that CNN lacked context relations. Therefore, to improve classification accuracy, some studies utilized a combination of CNN and Long Short-Term Memory (LSTM) [7, 8, 9].

Zhou et al. proposed a CNN-LSTM model [10], which leverages CNN to extract higher-level phrase representations and feeds them into an LSTM to obtain the sentence representation.

LSTM, CNN, and their hybrid counterparts have been successfully applied in a variety of natural language processing tasks, including sentiment analysis. Rehman et al. proposed an overly deep CNN-LSTM hybrid model [11], which includes dropout techniques, normalization techniques, and rectified linear units to

enhance the prediction accuracy. Although other studies have used similar hybrid approaches [12], the lack of an attention mechanism resulted in less improved results.

However, selecting the hyperparameters to train CNN models can become computationally expensive but is crucial for achieving optimal performance. Several works have addressed this issue and used optimization techniques such as random search [13]. Compared to grid search, random search provides several advantages, such as being able to add new trials to the experiment on the go, allowing changes in resolution, and stopping the experiment at any time [14, 15].

In this paper, we propose an optimized hybrid sentiment classification model that overcomes the limitations of the previous models. The experimental results show that our proposed model can effectively improve text classification accuracy.

3 PROPOSED ARCHITECTURE OF THE DOCUMENT CLASSIFICATION SYSTEM

3.1 Hybrid CNN and LSTM Model

The proposed architecture in this study is based on a hybrid model that combines Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks for text classification, where CNN is applied to extract the complicated features from the text and LSTM is exploited as a classifier. Figure 1 illustrates the structure of this model.

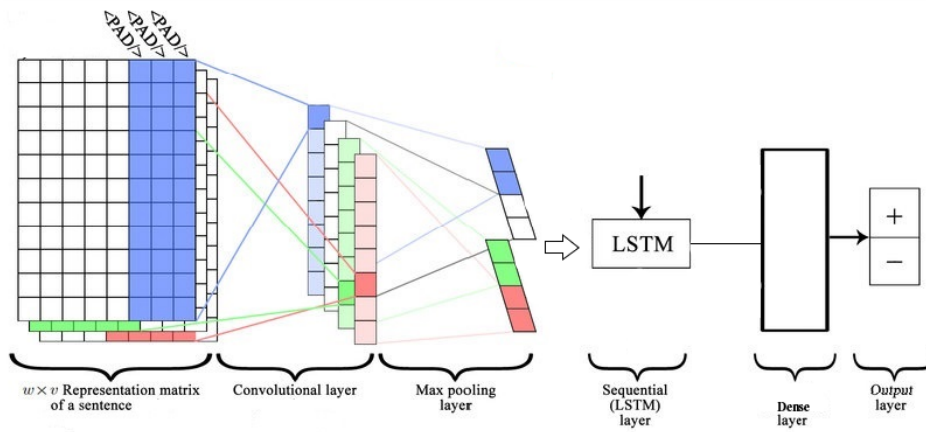


Figure 1. Hybrid model proposed architecture

The Input Layer is the first layer of the model, which consists of a fixed-dimension matrix of distinct vector embeddings representing each review as a row of vectors. Since each sentiment has different tokens, the tokens are based on the words in each review and are embedded with a nonidentical token.

The matrix size of this layer is $w \times v$, where v represents the length of the vector and w represents the number of tokens in the reviews. The maximum length of a review is defined as w, and any review with a lower number of tokens is padded to achieve the same length as the maximum.

The Convolutional Layer is applied to extract complicated features from the text by sliding the filter over the matrix to generate a new feature map. Various filters with different sizes are used to detect different features in the matrix. The filter strides or resorts only one column and one row over the matrix to detect multiple features in a review. An activation function is applied to define these features in the feature map.

The Max-Pooling Layer is utilized to down-sample the features in the feature map and compute the max value as a corresponding feature to a precise filter. The output vectors of this layer are then input to the LSTM networks to measure the long-term dependencies of feature sequences. The top value is selected in this step to attain the most significant feature and reduce the computation in the following layers.

The LSTM Layer is responsible for counting the anterior data and attaining sequential data. The output vectors of the previous layer are taken as inputs to this layer, which consists of a set number of units or cells. The closing vectors output of this layer are interconnected in one matrix in the range between 0 and 1 in the dense layer, and an activation function is used to classify the final output as either positive or negative.

4 TUNING HYPER PARAMETERS

Although this architecture combines CNN and LSTM networks to enhance text classification accuracy, it is computationally expensive to define the hyper-parameters for learning a CNN architecture and testing all the possible sets of hyper-parameters.

To optimize the hyper-parameters, the random search method has been widely used and has more benefits than grid search, as it allows practitioners to change the "resolution" on the go, add new trials to the set, or even ignore the failure test. Figure 2 illustrates a resumed idea about the Tuning method.

Hyper-parameters can be classified into two types [16]:

1. Network structure hyper-parameters, which include:
    - Training optimization algorithm – the method used to train the neural network by minimizing the cost function.
    - Network weight initialization – the process of setting the weights of a neural network to small random values that serve as the starting point for the optimization (learning or training) of the neural network model.
    - Hidden layers – the layers between the input and output layers.
    - Activation functions – mathematical functions that enable the model to learn nonlinear prediction boundaries.

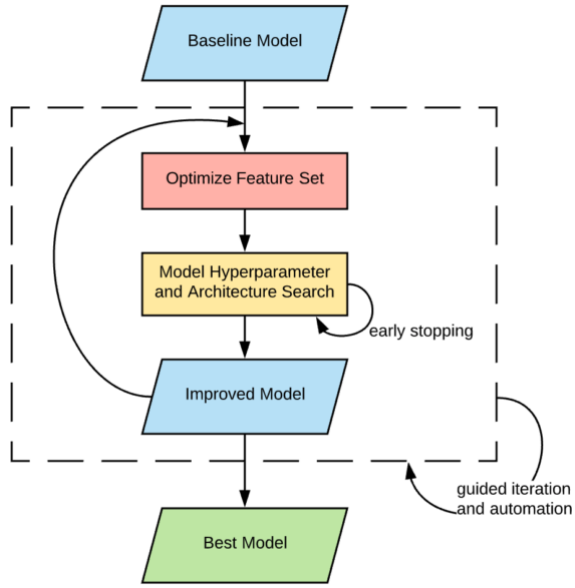


Figure 2. Tuning a model performance process

- Dropout regularization technique – a method for reducing over-fitting in artificial neural networks by preventing complex co-adaptations on training data.

2. Network training hyper-parameters, which include:

- Learning rate – the rate at which the weights are updated at the end of each batch.
- Momentum – a value that controls how much the previous update affects the current weight update.
- Number of epochs – the number of iterations of the entire training dataset to the network during training.
- Batch size – the number of patterns shown to the network before the weights are updated.

As the number of hyper-parameters can exceed 10, identifying the optimal combination can be seen as a search problem. To address this issue, an automatic optimizer, such as Random Search, can be utilized to achieve better results. The figure provided below depicts the process of hyperparameter tuning, which can be broken down into several steps.

The hyper-parameters of the model were trained based on the configurations outlined in Table 1, with values randomly assigned within their specified ranges. The

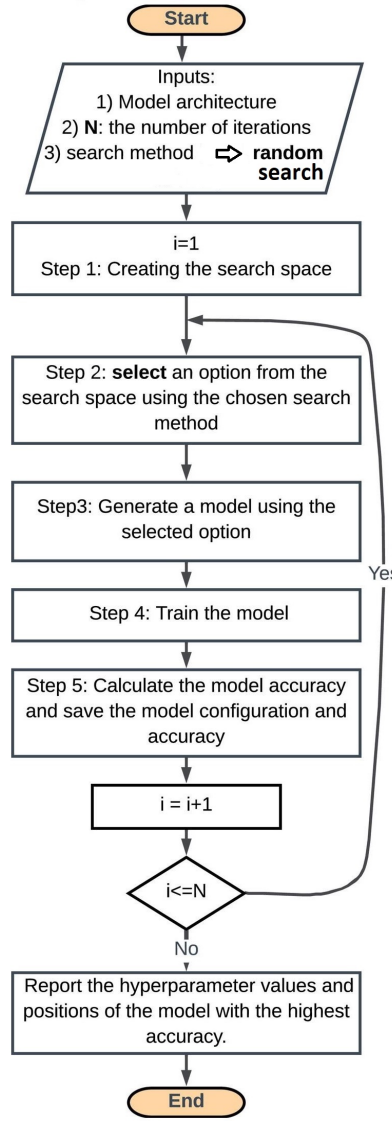


Figure 3. The hyper-parameters tuning process

evaluation of results was performed through multiple experiments. Table 1 presents the hyper-parameters that were considered for this study, with their corresponding ranges indicated within the square brackets.

Hyper-parameter Name	Hyper-parameter Value
Learning rate	[0.001, 0.01, 0.1]
Batch Size	[10, 20, 40, 60, 80]
Epochs	[3, 10, 50]
Momentum	[0.0, 0.2, 0.4, 0.6]
Optimizer	[SGD, RMSprop, Adagrad, Adadelta, Adam, Adamax, Nadam]
Init mode	[uniform, lecununiform, normal, zero, glorotnormal, glorotuniform, henormal]
Activation Function	[softmax, softplus, softsign, relu, tanh, sigmoid, hardsigmoid, linear]

Table 1. Hyper-parameters and their corresponding values

5 EXPERIMENTAL RESULTS

5.1 Dataset Description

To develop a precise classifier, obtaining an appropriate training dataset is critical, as it should encompass examples that accurately depict the outcomes targeted for prediction. In order to validate the reliability of our model, we conducted experiments utilizing the IMDB dataset for benchmark testing, which is described below.

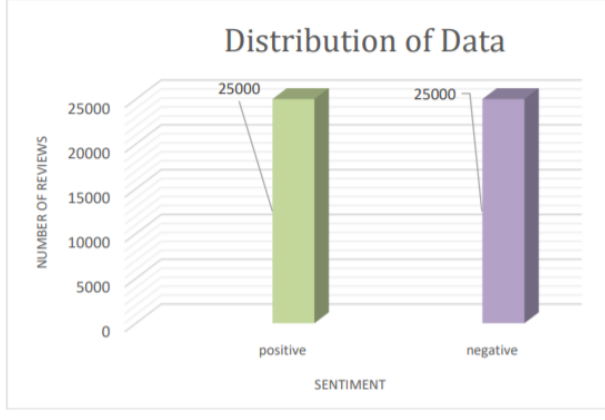


Figure 4. Dataset distribution

The IMDB dataset is a widely used resource in natural language processing and text analytics research. It consists of a large collection of movie reviews, totaling 50 000 in number. The primary objective of this dataset is to facilitate sentiment analysis tasks, which involve determining whether a given review expresses a positive or negative sentiment toward the movie being reviewed.

One of the standout features of the IMDB dataset is its size. Prior to the release of this dataset, benchmark datasets for sentiment analysis typically consisted of only a few thousand reviews. The IMDB dataset, on the other hand, contains a staggering 50 000 reviews, making it one of the largest publicly available datasets for this task.

The dataset is split into two equally sized sets of 25 000 reviews each, one for training and one for testing. This ensures that models developed using the dataset are evaluated on data that is independent of the data used for training, and helps to guard against overfitting. The reviews themselves are highly polar, meaning that they tend to express strong positive or negative sentiments toward the movies being reviewed. This makes the data-set well-suited for tasks such as binary sentiment classification, where the goal is to classify each review as either positive or negative.

In addition to its size and polarity, the IMDB dataset is also noteworthy for its diversity. The reviews cover a wide range of movies, spanning multiple genres, release years, and cultural contexts. This diversity helps to ensure that models developed using the dataset are able to generalize to a wide range of real-world scenarios, rather than being limited to a narrow subset of cases.

6 EVALUATION METHODS

The effectiveness of classifiers in discerning correct outcomes is typically assessed using well-established performance metrics, such as accuracy and loss rate. These measures are defined based on specific characteristics of the classification outcomes, which include:

True Positives (TP) – These refer to instances where the positive outcome is correctly predicted. For instance, when the actual class is positive, and the predicted class is also positive.

True Negatives (TN) – These instances occur when the negative outcome is correctly predicted. For example, when the actual class is negative, and the predicted class is also negative.

False Positives (FP) – These refer to instances where the negative outcome is wrongly predicted as positive. For instance, when the actual class is negative, but the predicted class is positive.

False Negatives (FN) – These refer to instances where the positive outcome is wrongly predicted as negative. For example, when the actual class is positive, but the predicted class is negative.

Accuracy – This is the most intuitive performance measure and is a ratio of the correctly predicted observations to the total observations. Accuracy is a valuable

measure because it provides an estimate of the extent to which the classifier is accurately predicting the outcome, hence its predictive power.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{1}$$

Loss – This is defined as the difference between the predicted and actual values. The most commonly used loss function in deep neural networks is cross-entropy, which is defined as the negative sum of the true value multiplied by the logarithm of the predicted probability. In mathematical terms:

$$CrossEntropy = -\sum_{i}\sum_{j} \log(p_{i,j}) y_{i,j}, \tag{2}$$

where $y_{i,j}$ represents the true value of sample i belonging to class j and $p_{i,j}$ is the predicted probability by the model that sample i belongs to class j.

Overall, the aforementioned evaluation methods enable the researcher to assess the performance of the classifier and determine whether it is effective in predicting the outcomes of interest.

7 RESULTS AND DISCUSSION

7.1 Construction of Environment and Parameter Setting

The use of a custom environment and parameter setting is essential in ensuring the reliability and reproducibility of our results. We recognize the importance of constructing a custom environment and parameter setting for our experiments.

To this end, we chose to use Python as our programming language and TensorFlow as our deep learning framework. Our lab environment, detailed in Table 2, showcases the hardware and software configuration we used in our study.

Despite the fact that our experiment was conducted on a hardware and software configuration with modest specifications, we were able to achieve significant and relevant results. This speaks to the efficiency and effectiveness of the custom environment and parameter settings we constructed, which were tailored to the specific research question we were addressing. Our findings demonstrate that it is possible to achieve meaningful results even with limited hardware resources, as long as the experimental setup is optimized appropriately.

To achieve our aim of sentiment analysis, we implemented a convolutional neural network (CNN) architecture that incorporated word embeddings of length 32 and permitted a maximum review length of 500. We trained the model using a batch size of 60 and applied a 32-filter kernel with a convolution layer kernel size of 3, utilizing ReLU as the activation function and sigmoid in the dense layer. Furthermore, we utilized an early stopping iteration function that minimized the loss value to attain

Software and Hardware	Configuration
CPU	Intel Core(TM) i7-7500UCPU, 2.70 GHz
RAM	8.00 GB
GPU	Intel HD Graphics 620
Operating System	Windows 10 Pro
Development Environment	Python 3.7, Jupyter Notebook

Table 2. Lab environment

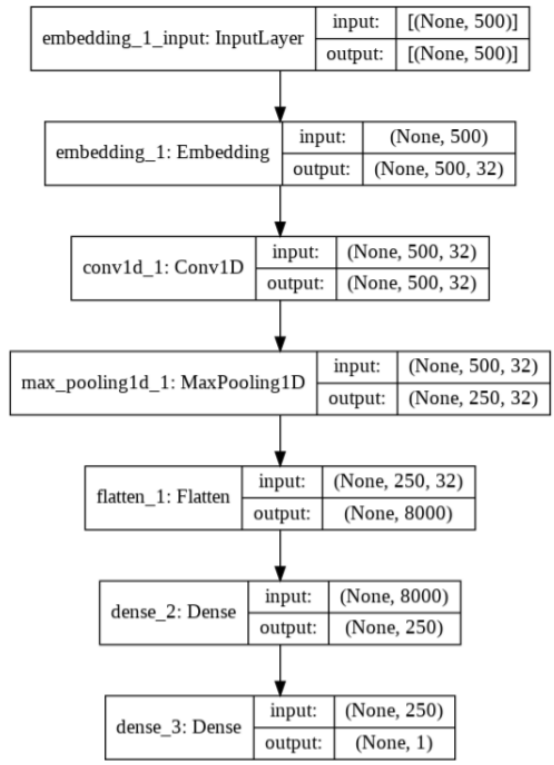


Figure 5. CNN model Layer's visualization

our final classification model. Our findings, illustrated in Figure 5, attest to the efficacy of this approach.

Regarding the CNN-LSTM model employed in our experiment, we utilized the following architecture: word embeddings vector length of 32 and a maximum review length of 500, a batch size of 60, and a convolution layer that applied a filter of 64 with a kernel size of 3 and utilized ReLU as an activation function. Additionally, the max pooling layer was set to a pool size of 2, and the dense layer used sigmoid activation. Finally, the early stopping iteration function was used to obtain the final

classification model with the loss value set to a minimum. This is demonstrated in Figure 6.

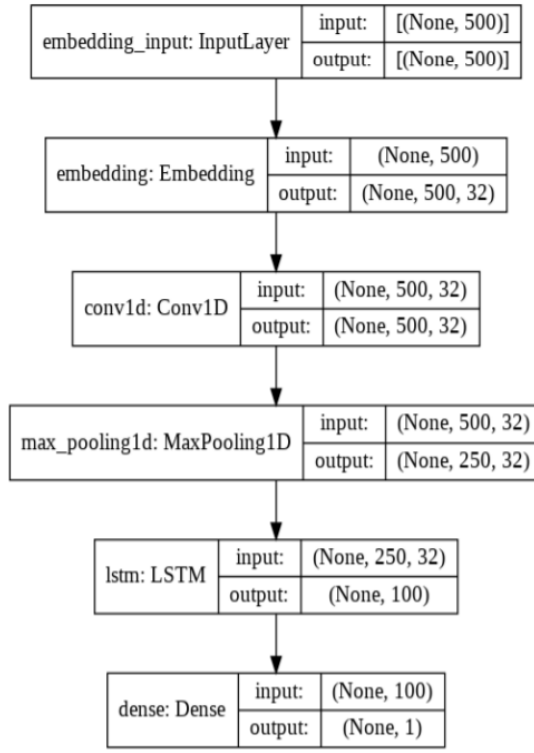


Figure 6. LSTM-CNN model Layer's visualization

The performance of the LSTM-CNN hybrid model was evaluated on both the validation and test sets, with classification accuracy, loss, and total execution time being used as metrics. The classification accuracy measures the percentage of correctly classified instances, while the loss function computes the difference between the predicted and actual values.

After conducting multiple executions of the CNN-LSTM model, accuracy was obtained, which is presented in Figure 7. However, to ensure that this accuracy was the optimal one, a random search tuning process was implemented. This involved varying the hyperparameters while keeping the architecture fixed, in order to explore new combinations. The hyper-parameters that were varied included learning rate, batch size, and the number of filters.

Figure 7 illustrates the best results achieved after numerous iterations and tuning adjustments.

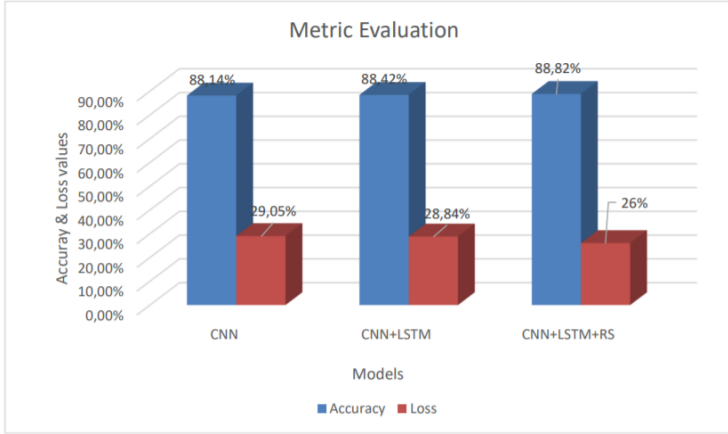


Figure 7. Metrics variation results on CNN and CNN-LSTM

It is evident that the LSTM model possesses a superior ability to capture contextual features in natural language compared to the standard CNN model when analyzing textual data.

This can be observed through the performance of the CNN-LSTM model, which surpasses the CNN model on both the test and validation sets, with an improvement of 0.28 % and 1.79 % respectively, indicating that the classification accuracy of the LSTM output is better. Consequently, it can be concluded that the performance of the CNN-LSTM model in the experiment is superior to that of the CNN model.

It is also noteworthy that the accuracy of the models was further improved by implementing random search hyper-parameter tuning, which resulted in a modest improvement of 0.40 %. Although this increase may appear insignificant, it is essential to recognize that it may be partially attributed to the fortuitous selection of hyper-parameters.

To assess the fitting appropriateness of the models, we present the training history of the CNN and CNN-LSTM models in Figures 8 and 9, respectively, which were captured by the History callback in Keras during training.

Based on the analysis of the accuracy plots, it appears that the model could benefit from further training, given that the trend for accuracy continues to rise over the last few epochs across all models.

Additionally, it can be inferred that the models have not reached a state of overfitting to the training dataset.

In contrast, the loss plot reveals that the model's performance is somewhat inconsistent across both the training and validation datasets. The parallel plots initially show a similar trend, but they begin to diverge as training progresses. This discrepancy in performance could explain why the training was stopped at an earlier epoch.

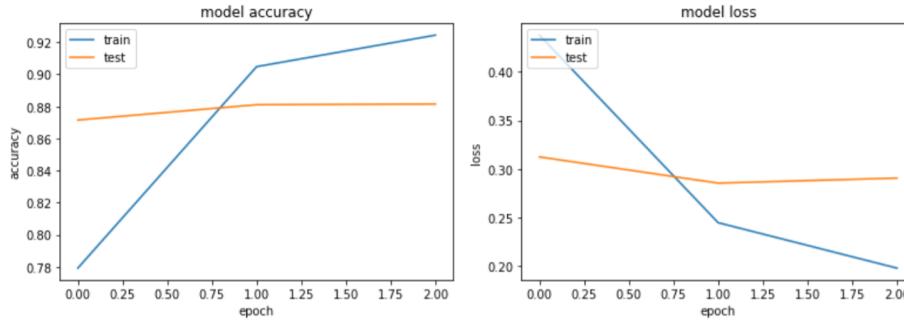


Figure 8. CNN Model training history

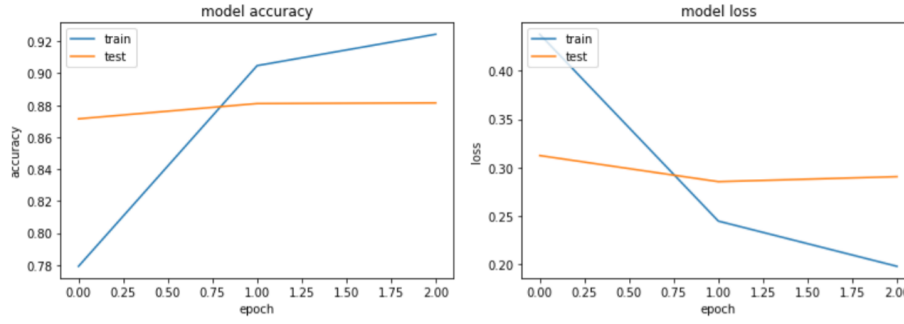


Figure 9. CNN-LSTM Model training history

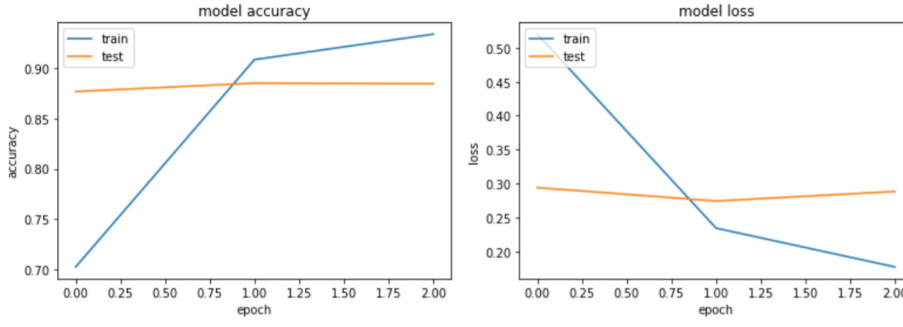


Figure 10. CNN-LSTM-RS Model training history

8 CONCLUSION

As the length and complexity of text increase, the task of text classification becomes more challenging. To address this issue, we propose a novel LSTM-CNN hybrid model for text classification. Our model combines the strengths of LSTM and CNN tasks to create a more effective deep-learning model. Our proposed model outperforms existing models in terms of accuracy and efficiency.

One advantage of our proposed model is that it can address the long-term dependency problem commonly encountered in existing models. Additionally, our model can mitigate the data loss problem, which is a common issue in traditional text classification models. To further improve our model's performance, we propose a method to tune its hyper-parameters using Random Search.

In future work, we plan to explore different architectures to incorporate into our model to further enhance its prediction performance. By continuously refining and optimizing our proposed LSTM-CNN hybrid model, we aim to provide a robust solution for text classification tasks, even for lengthy and complex texts.

REFERENCES

[1] Lakhotia, S.—Bresson, X.: An Experimental Comparison of Text Classification Techniques. 2018 International Conference on Cyberworlds (CW), 2018, pp. 58–65, doi: 10.1109/CW.2018.00022.

[2] Lewis, D. D.: Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval. In: Nédellec, C., Rouveirol, C. (Eds.): Machine Learning: ECML-98. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 1398, 1998, pp. 4–15, doi: 10.1007/BFb0026666.

[3] Suykens, J. A. K.—Vandewalle, J.: Least Squares Support Vector Machine Classifiers. Neural Processing Letters, Vol. 9, 1999, No. 3, pp. 293–300, doi: 10.1023/A:1018628609742.

[4] Safavian, S. R.—Landgrebe, D.: A Survey of Decision Tree Classifier Methodology. IEEE Transactions on Systems, Man, and Cybernetics, Vol. 21, 1991, No. 3, pp. 660–674, doi: 10.1109/21.97458.

[5] Cai, J.—Li, J.—Li, W.—Wang, J.: Deeplearning Model Used in Text Classification. 2018 15th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), 2018, pp. 123–126, doi: 10.1109/ICCWAMTIP.2018.8632592.

[6] Kim, Y.: Convolutional Neural Networks for Sentence Classification. CoRR, 2014, doi: 10.48550/arXiv.1408.5882.

[7] Zhou, P.—Qi, Z.—Zheng, S.—Xu, J.—Bao, H.—Xu, B.: Text Classification Improved by Integrating Bidirectional LSTM with Two-Dimensional Max Pooling. CoRR, 2016, doi: 10.48550/arXiv.1611.06639.

[8] Zhang, J.—Li, Y.—Tian, J.—Li, T.: LSTM-CNN Hybrid Model for Text Classification. 2018 IEEE 3rd Advanced Information Technology, Elec-

tronic and Automation Control Conference (IAEAC), 2018, pp. 1675–1680, doi: 10.1109/IAEAC.2018.8577620.

[9] Liang, D.—Zhang, Y.: AC-BLSTM: Asymmetric Convolutional Bidirectional LSTM Networks for Text Classification. CoRR, 2016, doi: 10.48550/arXiv.1611.01884.

[10] Zhou, C.—Sun, C.—Liu, Z.—Lau, F. C. M.: A C-LSTM Neural Network for Text Classification. CoRR, 2015, doi: 10.48550/arXiv.1511.08630.

[11] Rehman, A. U.—Malik, A. K.—Raza, B.—Ali, W.: A Hybrid CNN-LSTM Model for Improving Accuracy of Movie Reviews Sentiment Analysis. Multimedia Tools and Applications, Vol. 78, 2019, No. 18, pp. 26597–26613, doi: 10.1007/s11042-019-07788-7.

[12] She, X.—Zhang, D.: Text Classification Based on Hybrid CNN-LSTM Hybrid Model. Vol. 2, 2018, pp. 185–189, doi: 10.1109/ISCID.2018.10144.

[13] Bergstra, J.—Bengio, Y.: Random Search for Hyper-Parameter Optimization. Vol. 13, 2012, pp. 281–305.

[14] Bergstra, J.—Bardenet, R.—Bengio, Y.—Kégl, B.: Algorithms for Hyper-Parameter Optimization. In: Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., Weinberger, K. Q. (Eds.): Advances in Neural Information Processing Systems 24 (NIPS 2011). Curran Associates, Inc., 2011, pp. 2546–2554, https://proceedings.neurips.cc/paper_files/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf.

[15] Bergstra, J.—Yamins, D.—Cox, D. D.: Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. In: Dasgupta, S., McAllester, D. (Eds.): Proceedings of the 30th International Conference on Machine Learning (ICML). Proceedings of Machine Learning Research (PMLR), Vol. 28, 2013, pp. 115–123, http://proceedings.mlr.press/v28/bergstra13.pdf.

[16] Fan, X.—Runa, A.—Pei, Z.—Jiang, M.: An Improved Convolutional Neural Network for Text Classification. Journal of Physics: Conference Series, Vol. 2066, 2021, No. 1, Art. No. 012091, doi: 10.1088/1742-6596/2066/1/012091.

Chaima Ahle Touate is a doctoral candidate at the Faculty of Science and Technology, University Sultan Moulay Slimane. Specializing in artificial intelligence and text classification, she possesses a strong academic background with a Master's degree in business intelligence from the same institution. She is an active member of the Information Processing and Decision Support Laboratory TIAD. Her research interests revolve around data management, natural language processing, machine learning, and artificial intelligence, among others.

Rachid El Ayachi is an Esteemed Professor of higher education at the Faculty of Sciences and Techniques of Beni Mellal, specifically within the Computer Science Department. He has been serving in this position since 2013. He obtained his Master's degree in computer science, telecom and multimedia (ITM) from the Faculty of Sciences of Rabat, Mohammed V University, in 2006, and later went on to earn his Ph.D. in computer science from the Faculty of Sciences and Techniques of Beni Mellal, Sultan Moulay Slimane University, in 2012. Currently, he is an active member of the Information Processing and Decision Support Laboratory (TIAD). His research focuses on a wide range of areas including image processing, pattern recognition, machine learning, and natural language processing (NLP), among others.

Mohamed Biniz is a highly accomplished individual who completed his Master's degree in business intelligence in 2014 and his Ph.D. in computer science in 2018 at the Faculty of Science and Technology, University Sultan Moulay Sliman Beni Mellal. He currently holds the position of a Professor of computer science at the Polydisciplinary Faculty of the University Sultan Moulay Slimane Beni Mellal in Morocco. His research primarily centers around semantic web engineering and deep learning, with a specific interest in studying the evolution of ontology, Big Data, natural language processing, machine learning, and dynamic programming. His contributions in these fields have significantly advanced our understanding and application of cutting-edge technologies.

Computing and Informatics, Vol. 42, 2023, 667–689, doi: 10.31577/cai_2023_3_667

BERTDOM: PROTEIN DOMAIN BOUNDARY PREDICTION USING BERT

Ahmad Haseeb, Maryam Bashir, Aamir Wali

FAST School of Computing
National University of Computer and Emerging Sciences
Lahore, Pakistan
e-mail: l182081@lhr.nu.edu.pk, {maryam.bashir, aamir.wali}@nu.edu.pk

Abstract. The domains of a protein provide an insight on the functions that the protein can perform. Delineation of proteins using high-throughput experimental methods is difficult and a time-consuming task. Template-free and sequence-based computational methods that mainly rely on machine learning techniques can be used. However, some of the drawbacks of computational methods are low accuracy and their limitation in predicting different types of multi-domain proteins. Biological language modeling and deep learning techniques can be useful in such situations. In this study, we propose BERTDom for segmenting protein sequences. BERTDOM uses BERT for feature representation and stacked bi-directional long short term memory for classification. We pre-train BERT from scratch on a corpus of protein sequences obtained from UniProt knowledge base with reference clusters. For comparison, we also used two other deep learning architectures: LSTM and feed-forward neural networks. We also experimented with protein-to-vector (Pro2Vec) feature representation that uses word2vec to encode protein bio-words. For testing, three other bench-marked datasets were used. The experimental results on benchmarks datasets show that BERTDom produces the best F-score as compared to other template-based and template-free protein domain boundary prediction methods. Employing deep learning architectures can significantly improve domain boundary prediction. Furthermore, BERT used extensively in NLP for feature representation, has shown promising results when used for encoding bio-words. The code is available at https://github.com/maryam988/BERTDom-Code.

Keywords: Protein, protein domain boundary, BERT, biLSTM

1 INTRODUCTION

Protein domain boundary are the residues on a protein sequence where a domain starts and ends. A protein sequence or chain can consist of single domains or multiple domains where each domain is comprised of its own folded and independent sub-structures [1]. Protein domains are structural or functional units of a protein. Domains are recurring sequences that give very important information for the prediction of protein structure, function, and evolution. Numerous modular proteins families can have domains of different degrees of quantity and order [2]. Protein domains are building blocks of protein and so they can be arranged in different combinations to form proteins with more complex functions. Therefore, accurate identification of domains in protein is key to understanding the evolutionary mechanisms and protein function [3].

There are two ways of identifying domains in proteins: the first one is to predict boundaries of the domain from proteins having known three-dimensional (3D) structures, and the second one is the protein domain identification of those having unknown 3D structures. Domain boundary prediction is the first crucial step in protein classification and predicting protein 3D structures, which is a high-complexity problem [1]. Precise and accurate prediction of domain boundaries is the basis of various kinds of protein research because these researches start with the segmentation of a protein into its domains, which are its functional units [4]. The domain boundary prediction can optimize search methods for templates used in comparative modeling as the classification of templates is based on protein domains. Also, accurate prediction for homologous domains plays a central role in reliable MSA (multiple sequence alignment) [5].

Currently, the most accurate and reliable depiction of the protein domain is by experimental methods. Experimental methods for identifying protein domains require huge amount of proteins, effort and time. High-throughput technologies generate a large amount of data, so it is not possible to manually detect protein domain. This is why computational protein domain prediction methods are preferred. Computational methods use protein sequences to predict and identify protein domains. The delineation of protein domains using only protein sequences is still difficult. The computational domain boundary prediction methods mainly consist of template-based methods and ab-initio. Template-based methods use patterns or templates of existing similar protein sequences with known domain information for the prediction of proteins with unknown boundary information. Ab-initio methods use machine learning and statistical algorithms for prediction. These methods are more popular than template-based methods because they can be applied to any protein sequence. Some examples of these methods are DomPro [4], PPRODO [2], DROP [6], and DeepDom [3]. They are mostly used because they can predict any protein. However, the major drawback of ab-initio methods is low accuracy and precision as compared to template-based methods [3].

1.1 Motivation

Characterization of proteins using high-throughput experimental methods is difficult. Most of the template-free computational methods proposed for protein boundary domain prediction rely on machine learning techniques. To the best of our knowledge, not much work has been done to predict the domain boundary using deep learning methods except [3].

1.2 Objectives

The primary objective of this study is to predict the protein boundary domain using deep learning techniques. Furthermore, this study also aims to explore if deep learning techniques used in conjunction with biological language modeling and NLP techniques like bi-directional encoder representations from transformers (BERT) [7] can improve prediction.

Protein domain boundary prediction pipeline usually has the following steps. Protein sequences are segmented. For this purpose there are various techniques such as wordPiece, sentence-piece, or TAPE tokenizers like IUPAC and UniRef. Then the tokens or bio-words are encoded. BERT is a popular method for language representations. It provides a contextual representation of every bio-word in a sequence and can therefore be used for encoding. Other encoding schemes include word2vec and pro2vec. Finally, these representations are used to train classifiers. Thus, every step can be performed using a number of techniques. Another objective of this paper is to experiment with different combinations of segmentation-encoding-classification techniques and identify which combination works best for protein domain boundary classification. For this purpose, various deep learning architectures and methods like BERT, long short-term memory (LSTM) and fully convolutional neural networks (FCNN) are used which are extensively applied in other NLP and bio-informatics tasks. The prediction models are trained on protein sequences alone and does not rely on features engineering like sequence profile, solvent accessibility (SA), secondary structure (SS), etc.

1.3 Contributions

Following are the main contributions of this study.

- A protein domain boundary prediction model called BERTDom is proposed using deep learning techniques, BERTDom outperforms other template-based and computational techniques on benchmark datasets.
- Pre-trained BERT from scratch for protein bio-word embeddings for the first time.
- Protein vector representations created using pro2vec are used as features for protein domain boundary prediction.

- A multi-facet comparison is done involving two feature representations, three segmentation techniques and three deep learning models.

The rest of the paper is organized as follows. Section 2 presents necessary background required for understanding the problem of protein domain boundary prediction. Section 3 presents literature review on relevant work related to this study. Section 4 presents methodology used for protein domain boundary prediction in this study and Section 5 presents experimental details. Section 6 presents results and discussion and Section 7 concludes the study.

2 EXTENDED BACKGROUND

In this section, all the necessary concepts concerned with protein domain boundary are presented.

2.1 Protein and Its Domains

Protein performs a wide range of functions within living organisms, including transporting molecules from one location to another, responding to stimuli, providing structure to cells and organisms, DNA replication, and catalyzing metabolic reactions [8]. Protein is composed of amino acids and typically, 20 types of amino acids are found in proteins. Depending on the protein sequence, i.e., the position of amino acids in the protein chain, proteins fold into the specific 3D structure that allows them to do their functions and interact with other molecules and proteins. Proteins that have a common ancestor or diverged from the same ancestral gene are called homologous and have similar sequences [8].

Protein domain is a constant part of a protein sequence and makes a compact 3D structure that can fold independently. The length of a domain can be anywhere from 50 to 250 residues [9]. Due to molecular evolution, protein domains can be used as building blocks and they can be combined in different ways to form proteins with distinct structures and functions [10]. Each domain contributes to the overall functions of the protein. For instance, enzyme phospholipase D1 protein is a multi-domain protein since it has 3 different types of domains each performing a different sub-function to achieve an overall function of breaking down phosphatidylinositol.

Most domains comprise one continuous segment; some domains may consist of several discontinuous polypeptide segments [3]. The prediction and identification of discontinuous domains is still a very challenging problem.

2.2 Methods for Predicting Protein Domain Boundary

Methods for predicting protein domain boundaries are of two types: experimental and computational. These are discussed next.

2.3 Experimental Methods

Experimental methods are procedures performed on actual proteins. These methods use the particular biophysical or biochemical attributes of protein complexes. They can be done in a controlled lab environment (in-vitro) or inside a living organism (in-vivo). To speed up the process, high-throughput large-scale experimental methods have been designed to identify domains in a protein on a proteomic-wide scale. High-throughput experimental methods used for identifying domains are NMR (Nuclear Magnetic Resonance) analysis [9], and X-ray crystallography [11]. These methods are expensive in terms of labor, money, and time. These methods also need large quantities of proteins. Their results have high false negatives and false positives because the experiment's quality is affected by many factors [12]. Due to these limitations, computational methods are needed in the domain boundary prediction. Hence, it is of great practical importance to design accurate, reliable, and efficient computational methods to predict domains in less time, with high efficiency and at low cost.

2.4 Computational Methods

Computational methods for the prediction of protein domain boundary can be classified as template-based methods or template-free/ab-initio methods. The template-based methods search for similar protein sequences whose domain information is known and them map this information to the protein with unknown domain.

Some template-based approaches use sequence alignment in which the query and target protein sequences are aligned to predict the domain [13]. While other methods predict by aligning the secondary structure (SS) of a protein against the known domain boundary information of proteins given in class, architecture, topology, and homology (CATH) database [4].

Ab-initio or template-free methods are based only on the primary 1D protein sequence instead of any specific target protein [14]. These methods are more commonly used as compared to template-based methods since ab-initio methods can predict the domain boundary of any protein.

Ab-initio based machine learning (ML) methods directly or indirectly use the amino acid sequence as features to predict whether an amino acid is situated at a domain boundary. Ab-initio techniques are assisted by the accessibility of protein domain information databases. Ab-initio methods usually use the same input features like sequence profiles (SP), predicted solvent accessibility (SA) and predicted secondary structure (SS). For example, [15] also used amino acid composition and solvent accessibility to predict secondary structure.

The prediction accuracy of the ab-initio methods is usually lower than the template methods because of the lack of complete domain boundary information in protein sequence [3]. Most ab-initio methods are effective and successful in predicting domain boundaries when the target protein sequence has obvious resemblances to other sequences in domain classification databases or if the new

domains' length does not significantly differ from the average length of known protein sequences. In this paper, the focus is on identifying boundaries for proteins with two domains. The accuracy for one-domain proteins using computational methods is only 75–85 %, and it is significantly less for multi-domain proteins [5].

3 LITERATURE REVIEW

In this section, various computational methods for predicting protein domain boundaries are discussed. As mentioned earlier, computational methods can be ab-initio or template-based. A few hybrid techniques are also presented.

3.1 Template-Based Methods

Although the focus of this paper is the ab-initio methods, some template-based methods TBMs are briefly discussed.

Bondugula et al. [16] proposed FIEFDom, a homology-based approach for protein domain boundary prediction for multi-domain protein using features such as sequence profile and protein sequence using an FMO (fuzzy mean operator). The FMO assigns a likelihood score for each amino acid of the target sequence as corresponding to a domain boundary or not by using the NR (non-redundant) sequence database along with an RPS (reference protein set) database comprising already identified domain boundaries. This method vigorously identifies adjoining boundary sites. Authors claim the average prediction accuracy for single-domain and multi-domain proteins is 97 % and 58 % respectively. The proposed model has the ability to use new structure/sequence information after each RPS update without re-parameterization. When tested on other datasets having different domain information, this method consistently produced the same accuracy while other existing methods could not.

Zhidong Xue et al. [17] proposed another technique called ThreaDom, which infers protein domain boundary regions using multiple threading alignments. The key to this approach is that it can calibrate sequence alignment information and composite structure by generating a domain boundary profile from the multiple threading templates for exact domain prediction. ThreaDom correctly classifies 81 % of single-domain and multi-domain proteins when 78 % proteins have the domain linker allotted in the range of ± 20 residues. Finally, George et al. [18] developed SnapDRAGON, a 3D template-based approach for domain boundary prediction. It predicts domain boundary based on features from a secondary structure prediction and multiple alignments of protein sequences. SnapDRAGON utilizes the DRAGON method to generate a large set of alternative 3D models for a given multiple sequence alignment (MSA). Then it assigns domain boundaries automatically to each of the 3D model structures. Domain boundary assignment seen in the largest number of 3D models is selected. Model generation using this method leads to alternative 3D

model structures that differ in structure with associated boundary positions and have different domain contents. This technique used on NR dataset consisting of 414 multiple sequence alignments constitutes, 231 multiple-domain and 185 single protein chains registered an accuracy of 72.4 %.

3.2 Ab-Initio/ML-Based Methods

Sim et al. [2] proposed an ab-initio method for prediction of protein domain boundaries called PPRODO. PPRODO uses a feed-forward fully connected neural network with one hidden layer. A neural network is trained and tested for each residue in the protein sequence [19]. Amino acid residues in a protein sequence may mutate and this is more regular if the residues are close to domain boundaries. However, during the evolution some residues close to the domain boundaries may be conserved despite the usual movement of the domain. Analyzing the patterns in the position-specific scoring matrix can detect these features.

Cheng et al. [4] propose DOMpro that uses recursive neural networks to predict domain boundary using profiles, predicted secondary structure, and predicted relative solvent accessibility. This paper used the dataset from CATH database. The solvent accessibility and relative secondary structure are predicted for each sequence using ACCpro [20] and SSpro[21]. DOMPro can accurately predict the domain boundary and domains number for 25 % of the proteins that have two domains.

Yoo et al. [5] proposed the method DomNet that uses an enhanced general regression network (EGRN) specially created for managing high-dimensional protein sequences. DomNet uses a novel compact domain profile so that it can obtain more structural information efficiently from target sequences. The input features used by this method for training are predicted solvent accessibility information, predicted secondary structure, inter-domain linker index that detects the target protein sequence's possible domain boundaries and a compact domain profile. DomNet uses methods proposed by [22] for noise reduction, smoothing and searching vectors center by quantizing input vectors. DomNet reports the 71 % accuracy for proteins with multiple domains.

Ebina et al. [6] used a support vector machine (SVM) for prediction. This paper also used random forest to compute optimal input features which are then used to train SVM. Each amino acid residue is encoded into a 3000-dimensional vector. Various SVM classifiers were trained with different optimal feature candidate sets. SVM hyper-parameters were optimized using a SVMLab [23]. The proposed model named DROP, had sensitivity and precision values of 19.9 % greater than SVMs trained with non-optimized features using the same parameters. SVM was also used by Chakraborty et al. [24] based on input features composed of physiochemical properties of amino acids in protein sequence (obtained from AAIndex [25] database), predicted solvent accessibility and predicted secondary structure. Physiochemical properties of amino acid residues are linker index, hydrophobicity, linker propensity indices, polarity, and average flexibility indices. This method achieved

a precision, recall and accuracy of 0.79 %, 0.91 % and 78.58 % respectively, on the CASP10 dataset.

Eickholt et al. [14] developed DoBo, which also used SVM to classify the putative domain boundary signals. These signals are extracted from MSA generated by PSI-BLAST [17]. These MSA helps to detect assumed signals of domain boundary in a query protein by leveraging evolutionary information. MSAs often disclose the query protein's domain architecture by returning proteins comprised of domains analogous to the query protein sequence. DoBo has a recall and precision rate of 0.6. Finally, Bi-Qing Li et al. [26] also combined SVM with multiplefeature selection methods. This paper reported about 58–70 % higher specificity, 24–31 % greater MCC, and 28–40 % more accuracy than the DoMpro, Globplot, and Domcut methods but 20 % less sensitivity.

Hwan Hong et al. [1] proposed ConDo that used a 4-layer neural network for prediction of domain boundary. This method employed both short-range features such as sequence information, as well as long-range sequence information like evolutionary information and partially aligned sequences (PAS) in MSA. Long-range features are beneficial for deciding whether two residues belong to either separate domains or the same domain. Short-range features are residue position in a sequence, whether the residue is outside of the target chain, the number of residues in a sequence, sequence profile, predicted SA, and predicted SS. HHblits generates the sequence profile with UniRef20 database. SANN [27] predicted SA. PSIPRED [28] predicted SS. Neural networks' output layer has four units, which state whether or not the amino acid was within 20, 15, 10, or 5 amino acids from the correct domain boundary.

Jiang et al. [3] proposed DeepDom, a deep learning domain boundary prediction method that uses LSTM. DeepDom stacks multiple bi-directional LSTM layers to fit a non-linear high-order function with the aim of predicting the signal pattern of complex domain boundary. It uses a window sliding strategy to encode an input sequence into fixed-length protein fragments without considering the original length of the protein sequence. The majority of existing ab-initio domain boundary classifiers only permit users to provide and predict one protein sequence at one time. DeepDom does not perform the time-consuming and computationally intensive task of sequence profile generation method.

3.3 Hybrid Methods

Hybrid methods combine both ab-initio and template-based techniques. Walsh et al. [29] used bi-directional recurrent neural networks for predicting protein domain boundaries. The work also used structural classification of proteins (SCOP) and protein data bank (PDB) template profiles. Using template information improves the performance of ab-initio. Cheng et al. [30] describe DOMAC, a hybrid domain boundary prediction technique that integrates domain parsing, ab-initio, and homology modeling methods. This hybrid approach uses neural networks and the homology-based method to predict domain boundaries for proteins having homolo-

gous template structures in PDB to predict domain boundaries for new proteins.

4 METHODOLOGY

A protein sequence can be segmented using a number of techniques. The segmented bio-words can further get encoded using different techniques. For classification, a variety of deep learning models are available. In this section, we not only highlight the proposed BERTDom model, but also specify different combination of segmentation-encoding-classification techniques that were used in this paper for experimentation and comparison with BERTDom.

The high-level block diagram for BERTDom is given in Figure 1. In the first step, protein sequence is segmented into bio-words using wordPiece segmentation algorithm. Then every bio-word is encoded using BERT. Finally, the entire encoded protein sequence is fed to the stacked biLSTM classifier that predicts the domain boundary. Each of these step are discussed next. The state-of-the-art deep learning and NLP components of BERTDom model are also sufficiently discussed due to the multi-disciplinary nature of the current study.

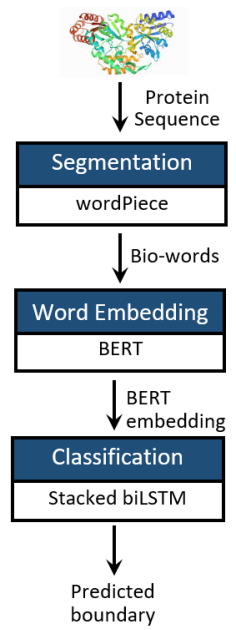


Figure 1. Architecture of BERTDom based on BERT and stacked biLSTMs for protein domain boundary prediction

4.1 WordPiece Algorithm for Bio-Word Segmentation

WordPiece algorithm [31] is a tokenizer that splits sentences into words and then words into sub-words. It is used in natural language processing (NLP) and deep learning architectures like BERT. WordPiece is trained for protein sequences and can therefore be used for segmenting protein sequence. The training parameters include the vocabulary size: 80 000, minimum words frequency: 2, and maximum sequence length: 256. WordPiece outputs a vocabulary file containing all words, sub-words, and individual characters in protein sequences. The trained wordPiece tokenizer is fed this vocabulary file along with the protein sequences and it segments them into bio-words.

4.2 Pre-Training of BERT Language Model for Protein Word Embeddings

Bidirectional encoder representations from transformers (BERT) is developed by Google AI researchers [7]. BERT consists of two steps: pre-training and fine-tuning. In pre-training, a large amount of unlabeled text is input to the BERT model for training where it learns the contextual relations between words and sentences in the language. BERT has two sub-models: masked language modeling (MLM) and next sentence prediction (NSP). MLM takes in a sentence with some masked words and it needs to predict the masked words. During fine-tuning, the last output layer of BERT is replaced by a new fully-connected layer that is trained for the specific task. Although each task is initialized with the same pre-trained weights in the non-final layers, the last layer of BERT is fine-tuned. The same pre-trained BERT weights can also be used to initialize other deep learning models for any sequence based prediction task. Protein domain boundary prediction can be modeled as a sequence based prediction task. This study proposes to use BERT for feature representations of protein sequences. Since there is no pre-trained BERT model available for protein bio-words, BERT had to be pre-trained for protein bio-word embeddings from scratch.

Figure 2 shows the architecture of one encoder state of BERT. BERT has multiple encoder states. Each encoder state has the same architecture. BERT processes sequence-based information by using a multihead attention mechanism. The input to BERT is a vector of words that have positional encoding information added to them. The self-attention layer takes the dot product of the input word with all query vectors of all other words in the sequence. A normalization layer is added after the self-attention layer. The next layer is the feed-forward neural network layer. The output of one encoder is passed as input to the next encoder state. The final output is the vector representation of input words such that the representation of each word has information of surrounding words baked into it.

The vocabulary file is converted to TFRecord format which is then used to pre-train BERT. The model configurations are given in Table 1. This BERT model's

configuration is the same as BERT-Medium uncased configuration – only difference is vocabulary size, which is changed from 30 500 to 80 000.

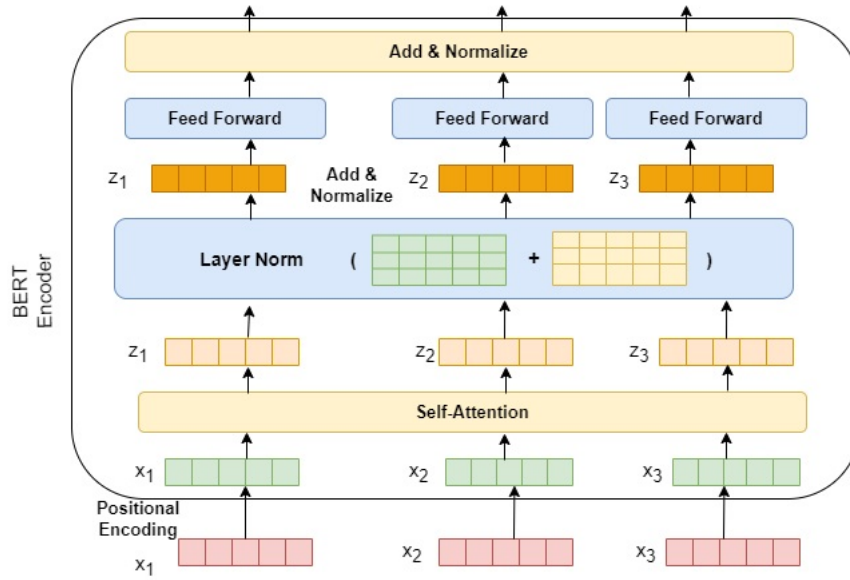


Figure 2. Architecture of BERT encoder [32]

Batch size of input examples	16
Maximum sequence length	256
Maximum predictions per sequence	35
Number of Training steps	30 000
Learning Rate	1e−4
Optimizer	Adam

Table 1. Hyper-parameters for pre-training BERT

The proteins can be of variable-length, so they are broken down into fixed-length protein sub-sequences using a sliding window strategy. The optimal values of window and stride were found to be 200 and 80 respectively. Each of these protein sub-sequences are tokenized using WordPiece tokenizer and then vectorized by the pre-trained BERT. BERT gives an embedding of 512 dimension for each token.

For fine-tuning BERT, BERTDom uses stacked biLSTM. However, we also performed experiments using 2 other deep learning techniques. These are also discussed in the following sub-sections.

4.3 BERTDom: Stacked Bi-LSTM and BERT's Language Model

For fine-tuning BERT, three models were used. The first model is stacked bidirectional LSTMs. LSTM [33] is a deep learning architecture that can process a sequence of data such as speech, text, or time-series. The true power of LSTMs lies in their ability to model longer sequences. LSTM is modified form of recurrent neural network (RNN) which were proposed for representation of sequence data. RNN suffer from the problem of vanishing gradient which occurs for long sequence of data. The gradient can become very small during back propagation in long sequences, this is called vanishing gradient problem [34]. LSTM overcome this problem by using a cell state for remembering only important information.

BiLSTM has one LSTM that processes sequence from start-to-end and another LSTM that processes the same sequence backwards. These two LSTMs are combined using the concatenation operator. Stacked biLSTM has multiple layers of biLSTM stacked n top of one another. The pre-trained BERT model is attached to stacked bidirectional LSTMs that has four bidirectional LSTM layers. The softmax is used as the activation function. The number of output units in last layer of each LSTM is equal to the maximum length of protein sequence which is 200.

The second deep learning model used for fine-tuning BERT is LSTM. The number of output units in the last layer is 200 with the softmax as the activation function.

Finally, the third deep learning model used for fine-tuning BERT is a deep feed-forward neural network. The network has 4 hidden layers with 1500 units and dropout values of 0.5, 0, 0, and 0.5 respectively. ReLU and sigmoid are used as activation functions. The number of output units in the last layer is equal to the length of the protein sequence – 200.

4.4 Feature Representation for Protein Bio-Word Using Protein-to-Vector (pro2vec)

In this study, for comparison purposes, we also used another feature representation method for segmented protein bio-words called pro-to-vector (pro2vec)[35] instead of BERT. For pro2vec model, word2vec algorithm called the skip-gram is used. Word2vec's skip-gram is used for learning the distributed representation for every protein word in proteins. We also trained word2vec from scratch on 185 000 protein sequences obtained from UniRef dataset [36] in sequence clusters with identity of 50 % (UniRef50). For classification, bidirectional LSTM is used. For segmentation, sentencePiece segmentation, [37] and K-mer segmentation techniques are used instead of wordPiece. The same window size and stride of 200 and 80 respectively, are used. Lastly, all protein word vectors of a protein sequence are combined together to form the embedding matrix of the protein sequence and then fed to a bidirectional LSTM for prediction of domain boundaries in a protein.

4.4.1 SentencePiece Segmentation

SentencePiece is a language independent tokenizer which is used when size of vocabulary is already known. It is trained directly from raw text using unigram language model (ULM). The sentencePiece library is used to implement this technique [38]. SentencePiece is an unsupervised method for tokenizing text.

4.4.2 K-mer Segmentation

K-mers are subsequences of length k in a protein sequence. For a given protein sequence, k-mer segmentation is used to divide them into bio-words. For example, the sequence MSLQ would have four monomers (M, S, L, and Q), three 2-mers (MS, SL, LQ), two 3-mers (MSL and SLQ) and one 4-mer (MSLQ). For length Z of a given protein sequence, we will get $Z - k + 1$ k-mers or bio-words.

4.5 Comparison with Other Methods

We have compared our proposed methods with existing template-based approaches such as Pfam [39], and FIEFDOM [16]. In addition to template-based methods we have also compared our proposed methods with statistical and machine learning approaches such as DomPro [4], PPRODO [2], and DROP [6]. DeepDom [3] is a recently proposed deep-learning-based method that uses LSTM for protein domain boundary prediction. DeepDom [3] has shown superior performance as compared to many template-based and statistical methods so we also compared our proposed methods with DeepDom.

5 EXPERIMENTAL SETUP

This section presents details of training and test data used in experiments. The evaluation measures are also discussed.

5.1 Training Dataset

For training, 46 000 domain boundary annotations of proteins from the CATH [40] version 4.2 database were collected. Uniprot database [41] is used for downloading corresponding sequences of these proteins. After downloading proteins, CD-HIT [42] tool is used to cluster similar proteins that meet the predefined 40 % similarity threshold. The representative protein sequences have sequence similarity less than 40 % with every other protein [43]. The similarity threshold (40 %) is used to make sure sufficient diverse data is available for the training of LSTM and BERT models.

5.2 Test Dataset

The proposed methods are tested on the proteins in the critical assessment of techniques for protein structure prediction (CASP) dataset which is a benchmark dataset. CASP protein domain prediction competition provided the annotations of domain boundaries of test proteins. Proteins in the training dataset that have at least 40 % similarity with any test proteins were removed from the training dataset. CASP provided three types of test datasets for bench-marking. These test datasets are listed below. Their details can be found in [3].

1. Free modeling (FM) target proteins from CASP 9.
2. Multi-domain proteins from CASP 9.
3. Discontinuous domain targets from CASP 8.

Dataset	# of Proteins	Single Domain	Multiple-Domain
Free Modeling	22	12	10
Multi-domain	14	0	14
Discontinuous domain	18	1	18

Table 2.

5.3 Dataset Used to Pre-Train Language Model – UniProt UniRef50

The UniRef50 protein dataset was used for pre-training language models (BERT and Word2vec) is obtained from UniProt Knowledge base (UniProtKB) with reference clusters (UniRef). UniRef gives clustered sequences' sets from the chosen UniParc records and UniProt. It removes protein sequences that are redundant and acquires whole coverage of the sequence space at 3 resolutions, which are UniRef50, UniRef90, and UniRef100. UniRef50 dataset was used for pre-training the language model. UniRef50 dataset contains 185 000 protein sequences.

5.4 Parameter Settings

DeepDom [3] is trained on 57 000 protein sequences while our proposed BERT model is trained on the dataset described above. Word2vec is trained on UniRef50 dataset with a window size of 10 and word vector dimension of 50 for ULM and K-Mer methods. The ULM is implemented using the sentencePiece library. It trains on UniRef dataset with a maximum vocabulary size of 50 000. Word2vec is also trained on a training dataset with a window size of 10.

5.5 Evaluation Measures

The proposed methodology is evaluated using benchmark classification evaluation measures, precision, recall, F-score, and accuracy. The formulas for these measures

are given as follows:

$$precision = \frac{TP}{TP + FP},$$

$$recall = \frac{TP}{(TP + FN)},$$

$$Accuracy = \frac{TP + TN}{FP + FN + TP + TN)},$$

$$F\text{-}score = \frac{2 * precision * recall}{precision + recall},$$

where, when a residue is predicted a domain boundary region, then it is checked if it is within ±20 residues of the actual domain boundary region. If yes, then it is a true positive (TP). If no, then it is a false positive (FP). When a residue is predicted outside the domain boundary region, then it is checked if it is within ±20 residues of the actual domain boundary region. If yes, then it is a false negative (FN). If no, then it is a true negative (TN).

6 RESULTS AND DISCUSSION

In this section, the results of the proposed methods are discussed for all datasets. Performance comparison between all methods is discussed in the following sections.

6.1 Performance on Free Modeling (FM) Targets

Table 3 presents the results of our proposed methods using free modeling (FM) targets from CASP9. Our proposed methods can be categorized into two main categories based on feature representation. The first is BERT encoder and the second is pro2vec. BERT is used as an encoder for protein sequences and then it is fine-tuned using three different deep learning models (LSTM, BiLSTM, and FCNN). BERTDom (BERT fined tuned with BiLSTM) performs best as compared to other models. The F-score is 0.58. Pro2vec is the second feature representation method in our experiments. Pro2vec was used with K-mer and unigram language model for segmenting a sequence into bio-words. Different values of k (3,4, and 5) have been tried, it is shown by results that 3-mer performs better than 4-mer and 5-mer segmentation. The F-score using pro2vec with 3-mer is also 0.58. The results of pro2vec with unigram model (0.57) are also close to pro2vec with 3-mer. The performance of BERT fined tuned with LSTM and FCNN is much inferior to BERT fine-tuned with biLSTM. The reason for this difference can be the bidirectional nature of biLSTM which takes into account the context from both directions.

	Accuracy	Precision	Recall	F-Score
BERT fine-tuned with LSTM	0.53	0.70	0.42	0.52
BERT fine-tuned with Stacked biLSTM (BERTDom)	0.74	0.74	0.47	**0.58**
BERT fine-tuned with Deep FCNN	0.71	0.69	0.43	0.53
Pro2Vec with 3-mer (biLSTM)	0.73	0.71	0.49	**0.58**
Pro2Vec with ULM (biLSTM)	0.76	0.84	0.43	0.57

Table 3. Comparison of proposed methods for FM dataset

	Accuracy	Precision	Recall	F-Score
BERT fine-tuned with LSTM	0.48	0.75	0.39	0.51
BERT fine-tuned with Stacked biLSTM (BERTDom)	0.76	0.82	0.45	0.58
BERT fine-tuned with Deep FCNN	0.74	0.79	0.38	0.51
Pro2Vec with 3-mer (biLSTM)	0.74	0.70	0.51	**0.59**
Pro2Vec with ULM (biLSTM)	0.76	0.84	0.41	0.55

Table 4. Comparison of proposed methods for multi-domain protein dataset

	Accuracy	Precision	Recall	F-Score
BERT fine-tuned with LSTM	0.50	0.75	0.43	**0.55**
BERT fine-tuned with stacked biLSTM (BERTDom)	0.70	0.82	0.33	0.47
BERT fine-tuned with Deep FCNN	0.70	0.81	0.32	0.46
Pro2Vec with 3-mer (biLSTM)	0.67	0.66	0.37	0.47
Pro2Vec with ULM (biLSTM)	0.68	0.79	0.28	0.41

Table 5. Comparison of proposed methods for DCD Dataset

		Precision	Recall	F-Score
Template based methods	Pfam [39]	0.32	0.49	0.39
	FIEFDOM [16]	0.23	0.18	0.2
Statistical and machine learning methods	DomPro [4]	0.50	0.18	0.26
	PPRODO [2]	0.33	0.49	0.39
	DROP [6]	0.43	0.18	0.25
	DeepDom [3]	0.89	0.41	0.56
Proposed methods	BERT fine-tuned with stacked biLSTM (BERTDom)	0.74	0.47	**0.58**
	Pro2Vec with 3-mer (biLSTM)	0.71	0.49	**0.58**

Table 6. Comparison of proposed method (BERT with stacked biLSTM) with other methods for FM dataset

		Precision	Recall	F-score
Template based	Pfam [39]	0.50	0.55	0.52
	FIEFDOM [16]	0.34	0.23	0.27
Statistical and machine learning methods	DomPro [4]	0.50	0.14	0.22
	PPRODO [2]	0.5	0.52	0.51
	DROP [6]	0.68	0.26	0.38
	DeepDom [3]	0.76	0.45	0.57
Proposed methods	BERT fine-tuned with stacked biLSTM (BERTDom)	0.82	0.45	0.58
	Pro2Vec with 3-mer (biLSTM)	0.7	0.51	**0.59**

Table 7. Comparison of proposed method (BERT with stacked biLSTM) with other methods for multi-domain dataset

6.2 Performance on Discontinuous Domain Targets (DCD)

Table 4 presents results on discontinuous domain targets. The results on this dataset are similar to results on the FM dataset. Pro2vec with 3-mer performs best with an F-score of 0.59, whereas, BERTDom has similar results with an F-score of 0.58. The rest of the models do not perform as well as these two models. Pro2vec, based on word2ec, learns a representation of bio-words based on the context. This contextual information helps in learning a better representation of the input data.

6.3 Performance on Multi-Domain Targets

Table 5 presents results using multi-Domain targets. BERTDom shows best results for multi-domain targets with an F-score of 0.55. This model has good precision as well as better recall as compared to other models. The rest of the models have good precision but low recall so the F-score of the rest of the models is less than BERT fine-tuned with LSTM. BERT fine-tuned with FCNN has inferior performance as compared to BERT fine-tuned with LSTM or biLSTM. The reason for this performance is the sequential nature of protein sequence data. LSTM and biLSTM are sequence-based models which remember context information.

6.4 Comparison with Other Methods

Table 6 and Table 7 present a comparison of our best performing proposed models (BERTDom and pro2vec with 3-mer) with existing work. Existing work can be divided into two categories. The first category is template-based methods. Our proposed method BERTDom outperforms template-based methods using F-score with a large margin. The F-score with BERTDom is 0.58 for FM dataset as shown in Table 6, whereas, Pfam [39] and FIEFDOM [16] have very low F-score of 0.39

Dataset			Precision	Recall	F-Score
FM	Template based methods	Pfam [39]	0.32	0.49	0.39
		FIEFDOM [16]	0.23	0.18	0.2
	Statistical and machine learning methods	DomPro [4]	0.50	0.18	0.26
		PPRODO [2]	0.33	0.49	0.39
		DROP [6]	0.43	0.18	0.25
		DeepDom [3]	0.89	0.41	0.56
	Proposed methods	BERT fine-tuned with stacked biLSTM (BERTDom)	0.74	0.47	**0.58**
		Pro2Vec with 3-mer (biLSTM)	0.71	0.49	**0.58**
Multi-domain	Template based	Pfam [39]	0.50	0.55	0.52
		FIEFDOM [16]	0.34	0.23	0.27
	Statistical and machine learning methods	DomPro [4]	0.50	0.14	0.22
		PPRODO [2]	0.5	0.52	0.51
		DROP [6]	0.68	0.26	0.38
		DeepDom [3]	0.76	0.45	0.57
	Proposed methods	BERT fine-tuned with stacked biLSTM (BERTDom)	0.82	0.45	0.58
		Pro2Vec with 3-mer (biLSTM)	0.7	0.51	**0.59**

Table 8. A Summary table for comparison of proposed method (BERT with stacked biLSTM) with other methods

and 0.2 respectively. Similarly, on the multi-domain dataset, our proposed methods have superior results as compared to template-based methods as shown in Table 7. We have also compared our proposed models with other statistical and machine learning models. Overall, our proposed models outperform the compared methods. DeepDom [3] performs best among the compared methods and our proposed models outperform DeepDom [3] as well. These results strengthen our belief that BERT and pro2vec give superior representations for protein sequences as compared to existing approaches. Table 8 presents results summary of comparison of our proposed methods with other methods. The best results are highlighted in bold. This table clearly shows the superior performance of our proposed deep learning methods for protein domain boundary prediction as compared to other approaches (template based, machine learning and statistical).

7 CONCLUSIONS

Protein domain boundary prediction is an important step in understanding the function of a protein. Most of the template-based methods have low accuracy so in recent years many computational approaches have been proposed for this problem. In this study, we have proposed a novel method BERTDom which trains the BERT model for the problem of protein domain boundary prediction. BERT is a popular model for the representation of text due to the sequential nature of the text. The protein sequence is also an example of sequence data so experimented with BERT for protein sequence data. The results are encouraging and show the potential of this multi-head attention-based model for protein sequence problems. The results are superior to many existing machine learning and template-based methods. We have also tried pro2vec for this problem. Pro2vec is inspired from word2vec for context-based words representation. The results with pro2vec are also superior as compared to exiting computational and template-based approaches.

The performance of deep learning models highly depends on the amount of training data. Google's BERT models are trained for at least 1 000 000 steps and are fed millions of documents, whereas we have trained the BERT model with only 10 000 steps and 185 000 sequences. The reason for the small training size is the lack of computational resources. Having said that, the results are promising. Thus, this study shows the potential of pre-trained BERT for protein domain boundary prediction even when trained on a small data. It is expected that if BERT is pre-trained with more data, the results can further improve.

REFERENCES

[1] Hong, S. H.—Joo, K.—Lee, J.: ConDo: Protein Domain Boundary Prediction Using Coevolutionary Information. Bioinformatics, Vol. 35, 2019, No. 14, pp. 2411–2417, doi: 10.1093/bioinformatics/bty973.

[2] Sim, J.—Kim, S. Y.—Lee, J.: PPRODO: Prediction of Protein Domain Boundaries Using Neural Networks. Proteins, Vol. 59, 2005, No. 3, pp. 627–632, doi: 10.1002/prot.20442.

[3] Jiang, Y.—Wang, D.—Xu, D.: DeepDom: Predicting Protein Domain Boundary from Sequence Alone Using Stacked Bidirectional LSTM. Biocomputing 2019: Proceedings of the Pacific Symposium, World Scientific, 2018, pp. 66–75, doi: 10.1142/9789813279827_0007.

[4] Cheng, J.—Sweredoski, M. J.—Baldi, P.: DOMpro: Protein Domain Prediction Using Profiles, Secondary Structure, Relative Solvent Accessibility, and Recursive Neural Networks. Data Mining and Knowledge Discovery, Vol. 13, 2006, No. 1, pp. 1–10, doi: 10.1007/s10618-005-0023-5.

[5] Yoo, P. D.—Sikder, A. R.—Taheri, J.—Zhou, B. B.—Zomaya, A. Y.: DomNet: Protein Domain Boundary Prediction Using Enhanced General Regression Net-

work and New Profiles. IEEE Transactions on NanoBioscience, Vol. 7, 2008, No. 2, pp. 172–181, doi: 10.1109/TNB.2008.2000747.

[6] Ebina, T.—Toh, H.—Kuroda, Y.: DROP: An SVM Domain Linker Predictor Trained with Optimal Features Selected by Random Forest. Bioinformatics, Vol. 27, 2011, No. 4, pp. 487–494, doi: 10.1093/bioinformatics/btq700.

[7] Devlin, J.—Chang, M. W.—Lee, K.—Toutanova, K.: BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In: Burstein, J., Doran, C., Solorio, T. (Eds.): Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2019). ACL, Vol. 1, 2019, pp. 4171–4186, doi: 10.18653/v1/N19-1423.

[8] Wang, Y.—Zhang, H.—Zhong, H.—Xue, Z.: Protein Domain Identification Methods and Online Resources. Computational and Structural Biotechnology Journal, Vol. 19, 2021, pp. 1145–1153, doi: 10.1016/j.csbj.2021.01.041.

[9] Folkers, G. E.—van Buuren, B. N. M.—Kaptein, R.: Expression Screening, Protein Purification and NMR Analysis of Human Protein Domains for Structural Genomics. Journal of Structural and Functional Genomics, Vol. 5, 2004, No. 1, pp. 119–131, doi: 10.1023/B:JSFG.0000029200.66197.0c.

[10] Wikipedia Contributors: Protein Domain. Wikipedia, The Free Encyclopedia, `https://en.wikipedia.org/wiki/Protein_domain` (Retrieved 2022-02-11).

[11] Brenner, S. E.: Target Selection for Structural Genomics. Nature Structural and Molecular Biology, Vol. 7, 2000, No. 11, pp. 967–969, doi: 10.1038/80747.

[12] Wang, Y.—You, Z. H.—Yang, S.—Li, X.—Jiang, T. H.—Zhou, X.: A High Efficient Biological Language Model for Predicting Protein–Protein Interactions. Cells, Vol. 8, 2019, No. 2, Art. No. 122, doi: 10.3390/cells8020122.

[13] Marchler-Bauer, A.—Anderson, J. B.—DeWeese-Scott, C.—Fedorova, N. D.—Geer, L. Y.—He, S.—Hurwitz, D. I. et al.: CDD: A Curated Entrez Database of Conserved Domain Alignments. Nucleic Acids Research, Vol. 31, 2003, No. 1, pp. 383–387, doi: 10.1093/nar/gkg087.

[14] Eickholt, J.—Deng, X.—Cheng, J.: DoBo: Protein Domain Boundary Prediction by Integrating Evolutionary Signals and Machine Learning. BMC Bioinformatics, Vol. 12, 2011, No. 1, Art. No. 43, doi: 10.1186/1471-2105-12-43.

[15] Liu, J.—Rost, B.: Sequence-Based Prediction of Protein Domains. Nucleic Acids Research, Vol. 32, 2004, No. 12, pp. 3522–3530, doi: 10.1093/nar/gkh684.

[16] Bondugula, R.—Lee, M. S.—Wallqvist, A.: FIEFDom: A Transparent Domain Boundary Recognition System Using a Fuzzy Mean Operator. Nucleic Acids Research, Vol. 37, 2009, No. 2, pp. 452–462, doi: 10.1093/nar/gkn944.

[17] Xue, Z.—Xu, D.—Wang, Y.—Zhang, Y.: ThreaDom: Extracting Protein Domain Boundary Information from Multiple Threading Alignments. Bioinformatics, Vol. 29, 2013, No. 13, pp. i247–i256, doi: 10.1093/bioinformatics/btt209.

[18] Chivian, D.—Kim, D. E.—Malmström, L.—Schonbrun, J.—Rohl, C. A.—Baker, D.: Prediction of Casp6 Structures Using Automated Robetta Protocols. Proteins: Structure, Function, and Bioinformatics, Vol. 61, 2005, No. S7, pp. 157–166, doi: 10.1002/prot.20733.

[19] Altschul, S. F.—Madden, T. L.—Schäffer, A. A.—Zhang, J.—Zhang, Z.—Miller, W.—Lipman, D. J.: Gapped BLAST and PSI-BLAST: A New Generation of Protein Database Search Programs. Nucleic Acids Research, Vol. 25, 1997, No. 17, pp. 3389–3402, doi: 10.1093/nar/25.17.3389.

[20] Pollastri, G.—Baldi, P.—Fariselli, P.—Casadio, R.: Prediction of Coordination Number and Relative Solvent Accessibility in Proteins. Proteins, Vol. 47, 2002, No. 2, pp. 142–153, doi: 10.1002/prot.10069.

[21] Baldi, P.—Pollastri, G.: The Principled Design of Large-Scale Recursive Neural Network Architectures – DAG-RNNs and the Protein Structure Prediction Problem. Journal of Machine Learning Research, Vol. 4, 2003, pp. 575–602, https://www.jmlr.org/papers/volume4/baldi03a/baldi03a.pdf.

[22] Yoo, P. D.—Sikder, A. R.—Zhou, B. B.—Zomaya, A. Y.: Improved General Regression Network for Protein Domain Boundary Prediction. BMC Bioinformatics, Vol. 9, Suppl. 1, 2008, doi: 10.1186/1471-2105-9-S1-S12.

[23] Joachims, T.: Making Large-Scale Support Vector Machine Learning Practical. Advances in Kernel Methods: Support Vector Learning, MIT Press, 1999, pp. 169–184.

[24] Chakraborty, S.—Das, S.—Chatterjee, P.: Prediction of Domain Boundaries in Protein Sequences Using Predicted Secondary Structure and Physicochemical Properties of Amino Acids. 2014 International Conference on Circuits, Power and Computing Technologies (ICCPCT-2014), IEEE, 2014, pp. 1022–1026, doi: 10.1109/ICCPCT.2014.7054913.

[25] Kawashima, S.—Ogata, H.—Kanehisa, M.: AAindex: Amino Acid Index Database. Nucleic Acids Research, Vol. 27, 1999, No. 1, pp. 368–369, doi: 10.1093/nar/27.1.368.

[26] Li, B. Q.—Hu, L. L.—Chen, L.—Feng, K. Y.—Cai, Y. D.—Chou, K. C.: Prediction of Protein Domain with mRMR Feature Selection and Analysis. PLoS ONE, Vol. 7, 2012, No. 6, Art. No. e39308, doi: 10.1371/journal.pone.0039308.

[27] Joo, K.—Lee, S. J.—Lee, J.: Sann: Solvent Accessibility Prediction of Proteins by Nearest Neighbor Method. Proteins, Vol. 80, 2012, No. 7, pp. 1791–1797, doi: 10.1002/prot.24074.

[28] McGuffin, L. J.—Bryson, K.—Jones, D. T.: The PSIPRED Protein Structure Prediction Server. Bioinformatics, Vol. 16, 2000, No. 4, pp. 404–405, doi: 10.1093/bioinformatics/16.4.404.

[29] Walsh, I.—Martin, A. J. M.—Mooney, C.—Rubagotti, E.—Vullo, A.—Pollastri, G.: Ab Initio and Homology Based Prediction of Protein Domains by Recursive Neural Networks. BMC Bioinformatics, Vol. 10, 2009, No. 1, Art. No. 195, doi: 10.1186/1471-2105-10-195.

[30] Cheng, J.: DOMAC: An Accurate, Hybrid Protein Domain Prediction Server. Nucleic Acids Research, Vol. 35, 2007, No. Suppl_2, pp. W354–W356, doi: 10.1093/nar/gkm390.

[31] Schuster, M.—Nakajima, K.: Japanese and Korean Voice Search. 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2012, pp. 5149–5152, doi: 10.1109/ICASSP.2012.6289079.

[32] Alammar, J.: The Illustrated BERT, ELMo, and Co. (How NLP Cracked Transfer

Learning). http://jalammar.github.io/illustrated-bert/ (Retrieved 2021-09-11).

[33] Hochreiter, S.—Schmidhuber, J.: Long Short-Term Memory. Neural Computation, Vol. 9, 1997, No. 8, pp. 1735–1780, doi: 10.1162/neco.1997.9.8.1735.

[34] Hochreiter, S.: The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, Vol. 6, 1998, No. 2, pp. 107–116, doi: 10.1142/S0218488598000094.

[35] Yi, H. C.—You, Z. H.—Cheng, L.—Zhou, X.—Jiang, T. H.—Li, X.—Wang, Y. B.: Learning Distributed Representations of RNA and Protein Sequences and Its Application for Predicting lncRNA-Protein Interactions. Computational and Structural Biotechnology Journal, Vol. 18, 2020, pp. 20–26, doi: 10.1016/j.csbj.2019.11.004.

[36] The UniProt Consortium: Protein Information Resource. European Bioinformatics Institute, SIB Swiss Institute of Bioinformatics, https://www.uniprot.org/uniref/?query=uniprot.

[37] Kudo, T.: Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates. In: Gurevych, I., Miyao, Y. (Eds.): Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). ACL, 2018, pp. 66–75, doi: 10.18653/v1/P18-1007.

[38] Kudo, T.—Richardson, J.: SentencePiece: A Simple and Language Independent Subword Tokenizer and Detokenizer for Neural Text Processing. In: Blanco, E., Lu, W. (Eds.): Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP 2018). ACL, 2018, pp. 66–71, doi: 10.18653/v1/D18-2012.

[39] Bateman, A.—Coin, L.—Durbin, R.—Finn, R. D.—Hollich, V.—Griffiths-Jones, S.—Khanna, A.—Marshall, M.—Moxon, S.—Sonnhammer, E. L. L.—Studholme, D. J.—Yeats, C.—Eddy, S. R.: The Pfam Protein Families Database. Nucleic Acids Research, Vol. 32, 2004, No. Suppl_1, pp. D138–D141, doi: 10.1093/nar/gkh121.

[40] Dawson, N. L.—Lewis, T. E.—Das, S.—Lees, J. G.—Lee, D.—Ashford, P.—Orengo, C. A.—Sillitoe, I.: CATH: An Expanded Resource to Predict Protein Function Through Structure and Sequence. Nucleic Acids Research, Vol. 45, 2017, No. D1, pp. D289–D295, doi: 10.1093/nar/gkw1098.

[41] The UniProt Consortium: UniProt: A Worldwide Hub of Protein Knowledge. Nucleic Acids Research, Vol. 47, 2019, No. D1, pp. D506–D515, doi: 10.1093/nar/gky1049.

[42] Fu, L.—Niu, B.—Zhu, Z.—Wu, S.—Li, W.: CD-HIT: Accelerated for Clustering the Next-Generation Sequencing Data. Bioinformatics, Vol. 28, 2012, No. 23, pp. 3150–3152, doi: 10.1093/bioinformatics/bts565.

[43] Wang, D.—Zeng, S.—Xu, C.—Qiu, W.—Liang, Y.—Joshi, T.—Xu, D.: MusiteDeep: A Deep-Learning Framework for General and Kinase-Specific Phosphorylation Site Prediction. Bioinformatics, Vol. 33, 2017, No. 24, pp. 3909–3916, doi: 10.1093/bioinformatics/btx496.

Ahmad Haseeb received his M.Sc. degree in computer science in 2021 from the National University of Computer and Emerging Sciences, Pakistan. His current research interests include machine learning, and natural language processing.

Maryam Bashir is an Assistant Professor of computer science at the National University of Computer and Emerging Sciences. She earned her doctorate in computer science from the Northeastern University in Boston, USA. She is recipient of prestigious Fulbright Scholarship for Ph.D. studies in the USA. Her research interests include information retrieval, natural language processing, and evolutionary algorithms.

Aamir Wali has been teaching at the Department of Computer Science, FAST-National University of Computer and Emerging Sciences since 2004. He has his Ph.D. in computer science from the same university. His areas of interest include font development, writing systems, machine learning, image processing, human-computer interaction and virtual/augmented reality.

Computing and Informatics, Vol. 42, 2023, 690–715, doi: 10.31577/cai_2023_3_690

MTREEILLUSTRATOR: A MIXED-INITIATIVE FRAMEWORK FOR VISUAL EXPLORATORY ANALYSIS OF MULTIDIMENSIONAL HIERARCHICAL DATA

Guijuan Wang

Information and Technology School, Computer Science and Technology School
Southwest University of Science and Technology
Mianyang 621010, China
e-mail: guijuanwang@swust.edu.cn

Yu Zhao

Institute of Rural Development, Shandong Academy of Social Sciences
Jinan 250002, China
e-mail: yuzhaosdass@foxmail.com

Boyou Tan, Zhong Wang, Jiansong Wang, Hao Guo

Computer Science and Technology School
Southwest University of Science and Technology
Mianyang 621010, China
e-mail: 905109256, 78339239, 1666938053, 2698896107@qq.com

Yadong Wu*

Computer Science and Engineering School
Sichuan University of Science and Engineering
Zigong 645002, China
e-mail: wyd028@163.com

* Corresponding author

Abstract. Multidimensional hierarchical (mTree) data are very common in daily life and scientific research. However, mTree data exploration is a laborious and time-consuming process due to its structural complexity and large dimension combination space. To address this problem, we present mTreeIllustrator, a mixed-initiative framework for exploratory analysis of multidimensional hierarchical data with faceted visualizations. First, we propose a recommendation pipeline for the automatic selection and visual representation of important subspaces of mTree data. Furthermore, we design a visual framework and an interaction schema to couple automatic recommendations with human specifications to facilitate progressive exploratory analysis. Comparative experiments and user studies demonstrate the usability and effectiveness of our framework.

Keywords: Multidimensional hierarchical data, visual exploratory analysis, visualization recommendation, faceted visualization

1 INTRODUCTION

Multidimensional hierarchical data are commonly seen in life and scientific research; examples include census data, enterprise organization data and biological structure data. We call a multidimensional hierarchical structure an mTree for brevity considering that a tree is the most distinctive graphical depiction of a hierarchical structure [1]. Because the widths and depths of different layers and branches vary widely, mTree data feature high structural complexity of structure and an immensely high-dimensional combination space, which makes the exploration of such data a challenging task [2]. Users must go through a tedious and time-consuming process to interactively check and refine the exploration process to search for the combinations that are interesting or useful [3]. Machine learning and visualization can be adopted to accelerate exploration. Machine learning is leveraged to recommend the most important subset to decrease the search space, and visualization is used to present complex data and structures with intuitive graphical representations. Instead of repeated manual iterations, the intelligent visualization recommender can ease the exploration process by suggesting both important data and graphical views for analysts to browse [4].

However, creating intelligent visualization recommender system for mTree data is not easy. It requires a high level of expertise in mTree data visualization. On the one hand, visual mTree data exploration involves both multidimensional information understanding and hierarchical structure perception. To present the knowledge contained in multiple dimensions, techniques that organize the multiple dimensions in one chart are available, such as radar charts, parallel coordinate plots (PCPs) [5] and scatter plot matrices, faceted visualization techniques that organize several simple charts together, where each chart encodes one facet can also be used, such as the small-multiple and multiple coordinate view (MCV) technique. To present hierarchical information, many different visualization methods have been developed. Schulz

maintains an online survey treeVis website [6], but determining which is the most suitable method for a given dataset can be a challenge [7]. In practice, mTree data exploration often requires bespoke [8, 2] to combine multidimensional and hierarchical information. For ordinary users without programming skills, a more automatic technique would be more feasible.

On the other hand, some automatic tools have been developed to reduce the technique threshold of visualization, including rule-based recommendation tools [9, 10] ranking mechanic-based tools [11, 10], machine learning-based tools [12, 11] mixed-initiative tools [13]. These tools are mainly designed for tabular data. They do not directly support multidimensional hierarchical data.

To bridge the gap in intelligent visual mTree data exploration, we propose an automatic pipeline and a visual analytic framework mTreeIllustrator. Considering the large combination space of mTree data, the mTreeIllustrator cannot cover all possible combinations and visual representations. Inspired by the mixed-initiative user interface paradigm that enables human to collaborate with the intelligent agents [14]. We integrate the auto-generated faceted visualization into the interactive human exploration, to inspire users to efficiently interpret the mTree data and update their exploration directions. The main contributions are as follows.

1. We propose a novel machine-learning powered pipeline for the workflow of automatic mTree data visualization. With it, the most important subspace of mTree data is automatically selected and encoded as faceted visualizations.
2. We design a mixed-initiative visual analytic framework to couple the intelligent visualization recommendations with user selections to support progressive mTree data exploration. The framework also enables users to refine the recommended visualizations and data subspace, and to visually compare mTree structures.
3. We demonstrate the usability and effectiveness of the proposed method and framework by the comparative performance experiments and user studies.

The rest of this paper is organized as follows: Section 2 discusses the related work. Section 3 describes the task and architecture. The proposed model is presented in Section 4. The visualization design of mTreeIllustrator is presented in Section 5. Section 6 provides a systematic evaluation. We conclude our work in Section 7.

2 RELATED WORKS

This section presents the research topics that are most relevant to our work, namely, mTree data visualization and visualization recommendation.

2.1 Multidimensional and Hierarchical Data Visualization

Compared with tabular data, mTree data are more complex in terms of both their structures and information organization patterns. Visualization plays an important

role in exploring complex data [15]. Researchers have introduced various visualization techniques to improve the efficiency of analyzing multidimensional and hierarchical data. TreeVersity [16, 15] explores the cyclical changes in each dimension of mTree data by introducing visualizations such as tables and time trend charts. The McVA system [17] designs multiple coordinate views by combining hierarchical bubble charts, PCP charts, word cloud charts, and radar plots to perform a comparative analysis of different countries and regions. Sakairi et al. [18] conducted a visual comparison analysis on the dosages of different products materials by combining hierarchical data with stacked plots. Li [1] developed a hierarchical data comparison system that supports the interactive exploration and analysis of hierarchical data and allows users to visualize data by selecting different hierarchical visual layout algorithms to understand the characteristics of the data. The MCT method [2] uses a combination of rectangular tree diagrams and PCP charts to assist with the exploration of multidimensional information in hierarchical structures. A rectangular tree diagram is used to encode a hierarchy, and the four edges of the diagram are used as the four axes of parallel coordinates. Limited by the edge count of a rectangle, it can visualize at most four dimensions. Zhou et al. [19] proposed a visualization method to uncover the relationships of multiple attributes. PCP charts and visual interaction techniques are used to assist the analysis process and can help data analyst visually analyse the relationships between multiple attributes and target variables. The relationships are encoded using the sunburst diagram. With this diagram, analysts can determine the overall attribute relationships at a glance. Although the above techniques contribute greatly to mTree data exploration, the techniques themselves are relatively complex and require users to have some visualization knowledge.

2.2 Visualization Recommendation

The goal of visualization recommendation is to automatically recommend suitable charts based on the data characteristics of the given data to lower the technical threshold of visualization and improve the efficiency of data exploration. A number of mechanisms have been proposed to assist with visualization recommendation, mainly including rule-based methods and machine learning-based approaches.

Rule-based visualization recommendation can be traced back to the APT tool [20], developed by Mackinlay in the 1980s; this tool can automatically design effective graphical representations of relational information (e.g., bar charts, scatter diagrams, and connection diagrams). The tool is implemented using synthetic algebra and graph design guidelines. Mackinlay considered graphical representations as sentences of a graphical language. A wide variety of designs can be systematically generated by using the composite algebra that makes up a small set of the original graphical languages. In 1994, Sage [21] extended APT with more properties and enhanced the user-oriented design. In 2007, ShowMe [22] extended automatic representation to charts tables (often called small multiple displays), where VizQL is based on the algebra used in APT, thus improving the algebra and enabling com-

pilation into a database query language. Recently, Voyager [3, 4] aggregated the knowledge derived from previous works using expressiveness and validity criteria to evaluate visual coding options; this method integrates manual selection and rule-based selection and enables users to engage in interactive browsing and refinement based on multiple recommendations. In 2019, Moritz et al. [10] proposed Draco, which develops hard constraints (e.g., the shape encoding channel cannot represent quantity values) and soft constraints (e.g., by default, the temporal field is mapped to the X-axis) based on common visual design guidelines and uses those rules to recommend charts. Nan et al. [23] defined a set of visual language rules based on data transformation, aggregation and visual mapping; summarized seven common visualization tasks; and then recommended visualization charts based on these rules and tasks.

With the expansion of machine learning, many creative works have been proposed for visual chart recommendations based on artificial intelligence. DeepEye [11] trained a recommendation model based on RankSVM. Given a dataset, the model can select valid charts based on the data characteristics and rank them to obtain the top-k options. Dibia et al. [24] proposed Data2Vis, a neural network-based translation model for automatically generating visualizations from a given dataset. In this approach, the visualization generation problem is formulated as a language translation problem, where the data specification is mapped to the visualization specification using the Vega-Lite declarative language [25]. Text-to-Viz [26] supports automatic infographics generation from natural language statements. VizML [12] considers chart recommendation as a prediction problem, where the model predicts the visual encoding of data for the given data column(s).

The above research demonstrates the effectiveness of recommendation-based approaches in data visualization. However, the existing work has mainly focused on tabular data. Compared to tabular data, hierarchical data are more complex and cannot be directly supported. To address this problem, we propose an automatic pipeline and visualization framework for the visual exploration of mTree data.

3 DESIGN REQUIREMENTS AND ARCHITECTURE

3.1 Design Requirements

Based on the research problem, we have identified the following design requirements that form our automatic pipeline and the visual analysis framework.

R1. Automatic dimension combination and selection: After obtaining new data, users typically need to repeatedly select and check different dimension combinations to obtain meaningful results. Such repetitive tasks should be improved by automation procedures.

R2. Automatic chart recommendation: The target users have little or no visualization knowledge, so the system should be able to automatically help the

user determine the appropriate visualization for a given dimension or dimension combination.

R3. Support for iterative dimensions and charts refinement: The recommendation provided by a machine learning model may not be optimal. Sometimes the users want to change dimensions or refine the visual coding of a chart to meet their expectations, for example, adding new dimensions or changing the color of a scatter plot.

R4. Support for interactive exploration and comparisons involving hierarchical data: The developed system can support visual explorations and comparisons of hierarchical data with different sizes and granularities, it allows users to select a branch of the input hierarchical data for data dimension exploration, and it supports the comparison of data from different branches.

R5. Support for exploration history tracking: Unlike tabular data, hierarchical data possess a more complex exploration path. Therefore, the design should track and visualize the users' exploration path so that users can clearly know where they are and how they arrived there at any time to lighten their memory burden.

3.2 The Architecture

As shown in Figure 1, the architecture of mTreeIllustrator consists of an automatic recommendation pipeline module (Figure 1, right) and an interactive visualization module (Figure 1, left). After a user uploads data via the graphical user interface, the data are sent to the automatic pipeline. The machine learning-enabled pipeline includes three seamlessly integrated models. First, the subspace importance assessment model evaluates the importance of each dimension of the given mTree data using the random forest (RF) algorithm and outputs the most important subspace to the visualization recommendation model. The recommendation model predicts the chart type for each valid dimension or dimension combination and passes these chart types to the rule-based chart encoding model. Last, the encoding model translates the subspace data and chart types into graphical charts and sends the visualizations back to the user interface. Then, users can interactively explore and prioritize their exploration based on the recommendation results. The UI also provides a set of intuitive visual designs to present the overall mTree structure and the exploration path to simplify the process of exploring complex tree structures.

4 AUTOMATIC PIPELINE

We propose an automatic pipeline to assist users in exploring mTree data. The automatic pipeline consists of a dimension importance evaluation model, a visualization recommendation model and a rule-based chart encoding model. Those models are seamlessly connected, take the user data as inputs, select the most important subspace of the data, and present the subspace visually.

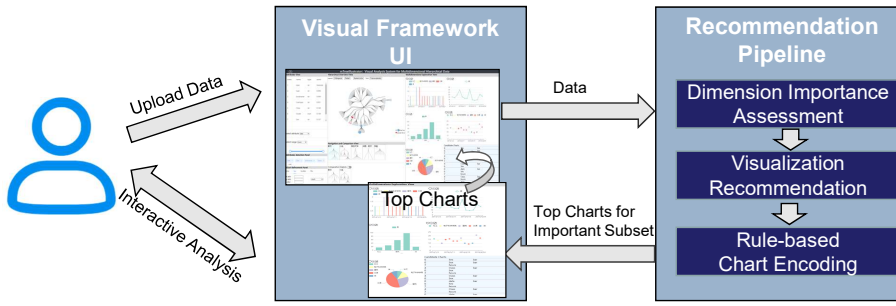


Figure 1. The architecture of mTreeIllustrator

4.1 Subspace Importance Evaluation Model

To understand mTree data, users need to iteratively check different dimensions and dimension combinations among layers and branches of the tree. With the numerous combinations of dimensions, layers and branches, the search space is large. Although many combinations are not important, users spend considerable time traversing them. To avoid wasting time on low-information dimensions or combinations, we propose a subspace importance assessment algorithm to allow the user to start their exploration from the most important combinations.

Several machine learning models are capable of subspace selection. Considering the interpretability, our model is designed based on the RF algorithm. The RF algorithm is used to select the subspace with the most important dimensions.

An RF comprises multiple tree sets (TSs). The majority of the tree decisions form the final decision. Each tree in a TS is a binary tree. The root node contains all training samples. According to a certain principle, each node selects the dimension that minimizes the "impurity" and uses this dimension as the branching dimension to split the node into two branches, each of which contains the corresponding sub samples. This process is repeated until the stopping condition is satisfied.

The frequently used measurements for "impurity" are the Gini index and out-of-bag (OOB) error. The accuracy of the Gini index is higher than that of the OOB when the signal-to-noise ratio is low, but in practice, it is difficult to obtain data with a low signal-to-noise ratio. The OOB error is more adaptive. Therefore, the OOB error is used to evaluate the importance of dimensions in our model. The OOB-based dimension importance measure is determined as follows: First, an RF is fit by applying the bootstrap aggregation (bagging) technique that repeatedly selects random samples with replacement and fits multiple trees based on these samples [27]. Then, to measure the importance of dimension X_i, in each tree, the OOB prediction error rate O_1 is calculated, the values of the dimension X_i are permuted among the training data, and the OOB error is computed again on the perturbed data set, namely O_2. Finally, the difference between O_1 and O_2 is calculated and normalized.

The difference on all TSs is calculated, and the average value is obtained, this value forms the importance score of X_i, which is denoted as $Vim_i^{(OOB)}$. Dimensions with larger values are ranked as more important than dimensions with smaller values. The $Vim_{ij}^{(OOB)}$ of dimension X_i in tree j can be calculated as follows:

$$Vim_{ij}^{(OOB)} = \frac{\sum_{p=1}^{n_o^j} I(Y_p = Y_p^j)}{n_o^j} - \frac{\sum_{p=1}^{n_o^j} I(Y_p = Y_{p,\pi}^j)}{n_o^j}, \tag{1}$$

where Y_p^j is the observed value of OOB in the j^{th} tree and $I(g)$ is the indicator function, which takes a value of 1 when the two values are equal and 0 when they are not equal. $Y_p \in \{0, 1\}$ is the result of the p^{th} observation, and $Y_{p,\pi}^j \in 0, 1$ is the predicted result of the p^{th} observation in the j^{th} tree after random replacement. When dimension Xi does not appear in the j^{th} tree, its importance is 0.

The importance of dimension X_i in the whole RF algorithm is calculated in (2), where n is the number of trees in RF.

$$Vim_i^{OOB} = \frac{\sum_{j=1}^{n} Vim_{ij}^{OOB}}{n}. \tag{2}$$

To calculate the dimension importance score for mTree dataset, the following steps are used.

Step 1. According to the size of the currently explored mTree data, the multidimensional data of each layer are merged to obtain a multidimensional set (MS).

Step 2. The dimensions in the MS are divided into a user set (US) and an evaluation set (ES). The US includes a user-focused dimension and has a size of one. The ES is the set of dimensions that are not selected by users. The aim of our model is to evaluate the importance of each dimension in the ES relative to the US. The higher the importance score is, the more significant the combination of it and the US is, and the more likely it can help users gain insights.

Step 3. The dimensions are ranked in descending order according to their importance scores. The top 3 important dimensions $\{I_1, I_2, I_3\}$ are returned.

Finally, the user focused dimension U and the top three related dimensions $\{I_1, I_2, I_3\}$ are chosen as the most important dimensions. Accordingly, the subset with dimensions $\{U, I_1, I_2, I_3\}$ of the selected branch or branches is returned as the important subspace.

4.2 LSTM-Based Subspace Visualization Recommendation Model

Selecting an appropriate visualization type for the important subspace is a complex task, and multifaceted information needs to be presented in a limited screen space. We propose an automated model to lower the threshold of this technique. Considering that the target users of our system have little visualization knowledge,

we specifically choose a less complex visualization technique: small multiples. It encodes multidimensional information with multiple simple charts, and each chart encodes one facet. We select four chart types, including bar charts, pie charts, line charts, and scatter plots, which are the most commonly used chart types for exploring multidimensional data [28]. These charts can help users complete the most frequent tasks, such as cluster analysis, correlation analysis, and anomaly detection [29].

Based on the design requirement, the recommendation process focuses more on the data exploration width. Therefore, the chart style, such as its color options, is beyond the recommendation scope. The aim of the recommendation model is to determine the suitable chart type for each valid dimension or dimension combination. Therefore, we formulate the recommendation problem as a classification problem: choosing one chart type from the four available types.

For recommendation model selection, two main modeling types are available: the learning-to-rank and the classification models. A learning-to-rank model is trained to judge whether one visual encoding is better than another; examples include the lambdaMART model, and the RankSVM model. A classification model such as Neural Network (NN) model, is used to predict the possible design choice. Based on the state-of-the-art research in visualization recommender systems [12, 10, 11], the NN based classification models have better precision. Furthermore, the long short-term memory (LSTM) model, a recurrent neural network (RNN) model variant, can overcome the vanishing gradient problem of traditional RNNs, and has been widely adopted in visual analysis frameworks in recent years [30, 31]. Therefore, in this work, we choose to adapt the LSTM model to predict chart types. The comparative experiments in the evaluation section (Section 6.1) demonstrate that it has better performance than the baseline NN and RankSVM models in our scenario.

The recommended workflow is shown in Figure 2. It starts from the incoming important subspace and formats it as a 4-dimensional table (1), it computes all valid combinations containing one to three dimensions (2), and for each combination, it extracts features (3) and sends them to the Bi-LSTM model (4) to predict the appropriate chart type (5).

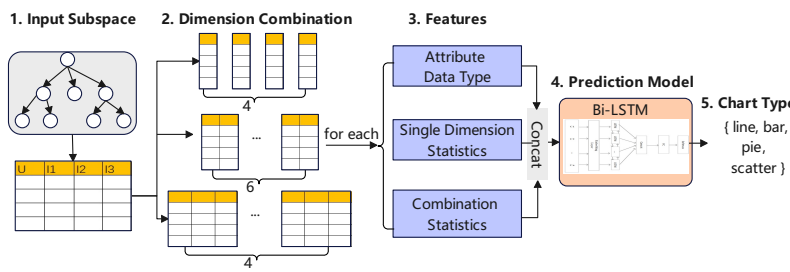


Figure 2. The workflow of visualization recommendation

The input of the recommendation model is a selected subspace with $\{U, I_1, I_2, I_3\}$, where U is the user selected dimension, and I_1, I_2, and I_3 are the top 3 important dimensions. The charts supported by our model can encode $1-3$ dimensions. Enumerating all possible cases with $1-3$ dimensions from $\{U, I_1, I_2, I_3\}$, we obtain $C_4^1 + C_4^1 + C_4^3 = 14$ combinations, some of which may not be valid. We obtain at most 14 valid dimension combinations. In turn, the visualization recommendation model will predict the most appropriate chart type for each valid combination.

Then, we need a way to convert those different characteristics into a multidimensional vector. Here, we refer to the approach in VizML [12] and the analysis in Table 1 and calculate the embedding vector of dimensions by feature engineering. The embedding vector consists of the type of dimension, the statistical characteristics of each single dimension (the total, mean, max, etc.), and the statistical characteristics of the dimension combinations.

Finally, the output layer uses the Softmax activation function to classify the input sequences and outputs the chart type with the highest probability.

4.3 Rule-Based Chart Encoding Model

The visualization recommendation model is only responsible for determining the chart type. We also need to determine how to map the $\{\text{dimension(s)}, \text{chartType}\}$ pair to a visual chart. For example, suppose that the input data contain two string-type dimensions $Mc1$ and $Mc2$, and that the recommended chart type is a bar chart. Then, a mapping rule is needed to determine which dimension is mapped to the X-axis and which is mapped to the Y-axis, as well as whether operations such as count and min are needed. In this example, the dimension with more categories should be mapped to the X-axis; suppose that this dimension is $Mc1$. Then, each value $Mc1_i$ in $Mc1$ is counted, and the percentage of each value $Mc2_i$ in the other dimension $Mc2$ is used as the color map of the bar chart. In addition, to avoid visual clustering, when the number of categories in $Mc2$ is greater than five, we select the four most frequent categories, and the rest of the categories are categorized as other.

We determine the rules with visualization expert interviews and refine the theme during practice. It would be better if a systematic study could be performed in the future. The mapping rules are developed and depended on the number of dimensions, the types of the dimensions and the chart characteristics. Many combinations of dimensions can be formed, and Table 1 lists only part of the encoding rules. In the table, S refers to the string-data type, N refers to the numeric type, and D refers to the temporal type.

5 VISUAL DESIGN

We design an interactive visualization framework, mTreeIllustrator, to fulfill the design requirements mentioned in Section 3.1. mTreeIllustrator mainly consists of

Dimension(s)	Recommended Chart	Rules
$\{S\}$	Pie chart	Count and compute the percentage of each category
$\{S, N\}$	Bar chart	Encode S as the X-axis, sum N based on S_i
$\{D, S, N\}$	Line Chart	When D is more than the threshold, map D to the X-axis, map N to the Y-axis, and use S for coloring. When D is less than the threshold, map S to the X-axis, map N to the Y-axis, and use D for coloring.

Table 1. Chart encoding rules

eight components (Figure 3): control panel (Figure 3 A–C), a hierarchical overview (Figure 3 D), navigation and comparison views (Figure 3 E–F), and multidimensional exploration view (Figure 3 G–H). These views coordinate with each other to allow users to conduct deeper exploration and comparison with the inspiration provided by the recommended visualizations.

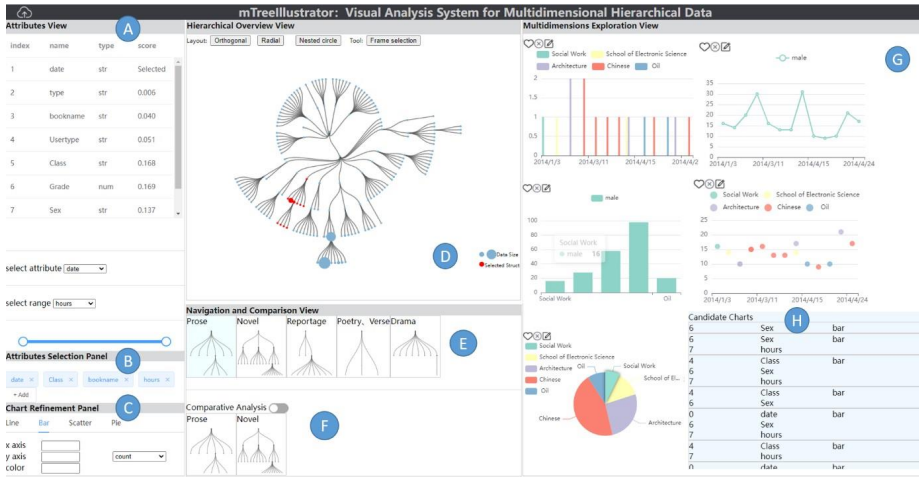


Figure 3. The interface of mTreeIllustrator. The left part contains an attribute view (A), an attribute selection panel (B) and a chart refinement panel (C); the middle shows the hierarchical overview (D) and the navigation and comparison views (E, F); the right part contains multidimensional exploration views, including the top 5 recommendation charts (G) and candidate charts (H).

5.1 Hierarchical Overview

The mantra "overview first, zoom and filter, then details on demand" [32] has been widely used in the design of complex data exploration systems. Thus, we follow this mantra and put the hierarchical overview view (Figure 3 D) in the center and surround it with the detailed views.

The hierarchical overview presents the overall structure and distribution of the uploaded hierarchical data (R3). Considering the scale and topological variance of user data, the system provides three layout methods (the top-left buttons) to allow users to switch layouts to better display their data. Each layout method has its own advantages. The orthogonal node-link diagram performs better in presenting structures, but its spatial utilization is low, and it is not suitable for large data. The radial node-link layout has better spatial utilization, but its presentation of the tree depth is limited because its root node is fixed to the center of the circle. Therefore, it is suitable for presenting compact hierarchical data with a small depth. The circular treemap is not as intuitive as the node-link diagram, but it can encode more data elements within the same screen space. It also has advantages in terms of internode comparisons among large hierarchical data [33].

After becoming familiar with the overall information, users generally want to perform deeper exploration based on their analysis interests (R4). To support this requirement, a box selection button (the top-right button) is designed to allow users to select a node or a branch for deeper fine-grained exploration.

5.2 Multiple Dimensions Exploration View

As shown in Figure 3 G, the multiple dimensions exploration view visually presents the recommendation result from the automatic recommendation pipeline (R3). A series of small charts are generated by the visualization recommendation model, and each chart presents a facet of an important subspace. The chart order is sorted according to their importance scores. Based on the recommendation pipeline described in Section 4, we obtain at most 14 graphical charts. These charts are ranked based on whether they encode the user selected dimension (U1), the dimension count, and the dimension importance scores. To promote exploration broadness, we present multiple charts based on the current user selection. However, to avoid overwhelming users with too much information, we need to limit the charts counts. Following the "the seven plus or minus two" rule proposed by psychologist George Miller [34], human short-term memory can store only five to nine pieces of information, five for complex information, and nine for simple information. Therefore, to strike a balance, only the top five charts are shown. The other candidates are listed in a table next to the top charts (Figure 3 H). If users are interested in a candidate chart in the table, they can click on it. The system will display the chart.

5.3 Navigation and Comparison Views

During exploration, another challenge is that the exploration path may be long due to the structural complexity of hierarchical data. To lighten users' memory burdens, the exploration history view (R4-5) is designed to track users' exploration path so that the users can clearly see where they are and how they got there at any time. The branches or nodes that a user has visited during exploration are saved as thumbnails based on the access order. The most recently visited data are inserted from the left. Furthermore, users often need to perform comparisons between different hops of the exploration history, and the comparison view (Figure 3 F) is designed to allow users to select a comparison target to compare (R5). By clicking on the history thumbnail or by directly selecting a branch from the hierarchical overview, a comparison target is selected. With the target, the back end of our system temporarily generates a classification dimension and treats it as a user-selected dimension. Then, the recommendation pipeline automatically generates the most relevant dimensions regarding this target dimension and refreshes the top charts in the multidimensional exploration view. This process can help users efficiently complete the multidimensional substructure comparison.

5.4 Control Panel

The control panels are designed to support users' deeper analyse and free exploration (R3-4), and they mainly consist of the dimension view (Figure 3 A), a dimension selection panel (Figure 3 B) and a chart refinement panel (Figure 3 C). Please note that in the UI design, the term "Attribute" indicates the "Dimension" in the recommendation pipeline. We use this term because it is easier for target users to understand. The attribute view presents the attribute name, attribute category, and importance score in the current exploration. The importance score is calculated by the subspace importance assessment model based on the user-selected attribute. In the attribute selection panel, users can choose attributes, and then the system passes the user selection to the recommendation pipeline. The goal of the chart refinement panel is to allow users to modify and refine the recommended charts (R4). Inspired by the design of Voyager, the panel mainly provides three functions: the chart type selection, data operation selection and visual coding. According to the recommendation pipeline, scatter plots, line charts, bar charts, and pie charts are supported. Tha data operations refer to the max, min, count, sum, and range calculations. For example, if the final chart is a scatter plot, the user can perform data operation on the y-axis. If the max operation is selected, the y-axis encodes the maximum of the data. The visual encoding editor supports the user in changing the element colors and sizes. Users can change the encoding setting according to their preferences. To edit a recommended chart in Figure 3 G, the user clicks on the chart, and then the system automatically loads the configuration options of that chart into the chart editor.

5.5 Interaction Design

Rich interactions are provided in the mTreeIllustrator user interface to facilitate the mixed-initiative data exploration. As shown in Figure 3, users can click to select their branch of interest in the mTree Overview chart (Figure 3 D), or set their desired attribute in the attribute view (Figure 3 A). Accordingly, the recommendation pipeline is automatically triggered to calculate the top attributes and visualizations related to the latest user selection, and then update the attribute table view (Figure 3 A) and the multidimensional exploration view (Figure 3 E). In addition, the system allows users to refine the system recommendation. They can add or delete the recommended attributes (Figure 3 B), change the chart encoding (Figure 3 C), or zoom out the candidate charts (Figure 3 H).

5.6 Scalability Consideration

For scalability, the current mTreeIllustrator design is targeted for moderate-size data that can be rendered in acceptable time and fit into the available screen size, i.e., thousands of data items. In exploration cases with larger data sizes, techniques such as Level-of-Detail (LoD) rendering may be extended from our framework.

6 EVALUATION

To verify the effectiveness and usability of the visualization framework proposed in this paper, we performed both performance evaluations and user studies.

6.1 Model Performance Evaluations

We conduct a comparative experiment to evaluate the performance of our model.

Data: The experimental dataset is derived from a subset of the VizML corpus which includes data and visual chart mapping pairs published by Plotly community users. After performing data cleaning, the valid dataset consists of 31 829 scatter plots, 12 002 bar charts, 23 702 line charts and 3 144 pie charts. For model training, the dataset is split into training/validation/test sets with a ratio of 60/20/20. The chart type distribution of this dataset is imbalanced, which may cause the prediction to be inclined toward the class with more samples and affect the generalization ability of the model. To prevent this problem, the class reorganization method [35] is used to balance the training dataset. This method needs to be repeated before each training step. The procedure is shown in Figure 4.

First, the original samples are classified and arranged by the chart type. Suppose that chart type M has the maximum number of samples. A random

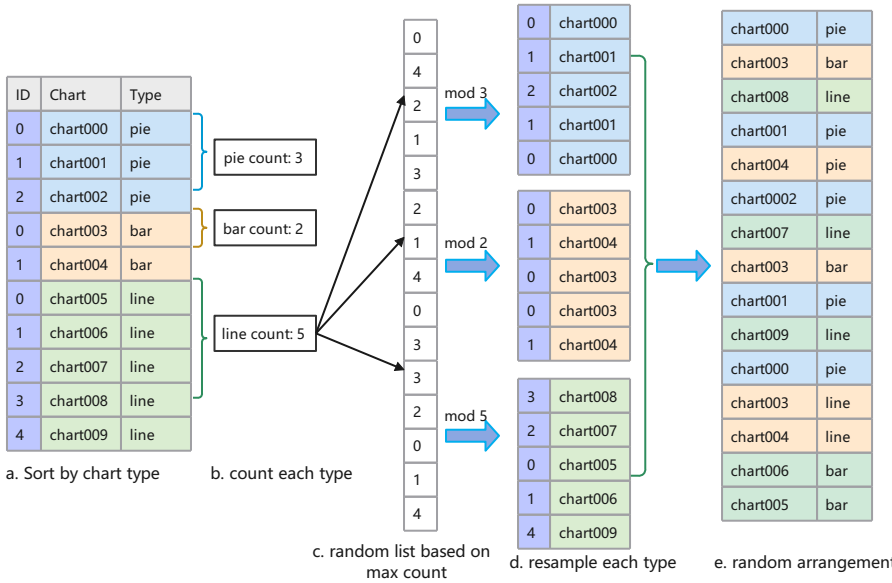


Figure 4. Procedure for balancing the training dataset

list L is generated for each class based on the count of M, and the random number in L is used to balance the number of samples in each class to obtain the corresponding index. Then, a random chart list (CLs) are generated by extracting charts from each class according to the index. All CLs are concatenated and randomly placed to obtain the last chart list (LCL). Now, the samples in each class in the LCL are equal. The advantage of this method is that it does not require extra information and can be run automatically.

Environment and Configuration: Our model is implemented with Python version 3.7 and PyTorch framework version 1.7.1 on a Windows desktop (Intel Core@2.30 GHz CPU with 12 GB of memory). The initial learning rate is set to $5 \times 10 - 4$, and the loss is reduced by a factor of 10 if the loss plateau is encountered; otherwise, the reduction is triggered every 5 interactions. The dropout rate is set to 0.5, the batch size is set to 128, and 100 epochs are run to train the model.

Procedure: We select three models which are used in the recent visualization recommendation systems as comparison, namely a support vector machine (SVM), a neural network (NN), and the RankSVM model. RankSVM is the model used by the DeepEyes visualization recommendation, and NN is used in the VizML and LQ2 tools. Among them, the NN has the best performance and is used as the baseline model. The evaluation metrics are as follows: the accuracy (Acc) is

calculated in (3); the precision (Pre) is calculated in (4); the recall (Rec) is calculated in (5); and the F1 score is calculated in (6). The metrics are computed based on the confusion matrix that counts the correct and incorrect prediction counts: TPs (true positive), FPs (false positive), TNs (true negative), and FNs (false negative), where $TP + FP + TN + FN =$ the total samples. The details are as follows:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}, \tag{3}$$

$$Pre = \frac{TP}{TP + FP}, \tag{4}$$

$$Rec = \frac{TP}{TP + FN}. \tag{5}$$

Results: The experimental results are given in Table 2. The evaluation metrics, Acc, Rec, Pre and F1 of our model are above 93.9 %, which is better than those of the other models. The results in the table show that the NN model outperforms the SVM model in terms of accuracy and F1 scores. Among the NN models, the recurrent RNN-based model (Ours) is slightly better than the baseline NN model, which may be because the RNN model better captures the data features during training and requires fewer samples.

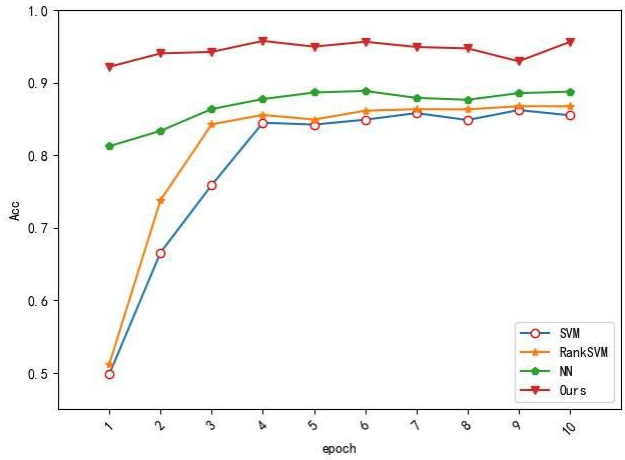
Model	**Acc**	**Rec**	**Pre**	**F1**
SVM	0.851	0.841	0.832	0.836
RankSVM	0.861	0.842	0.835	0.838
NN	0.881	0.874	0.863	0.868
Ours	0.949	0.946	0.939	0.942

Table 2. Performance metric distribution of the four models

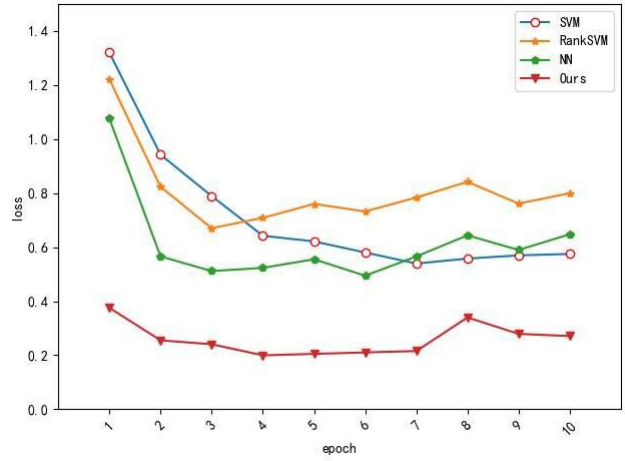
Ten epochs are run to compare the training time changes exhibited by the models, the accuracy (Acc) and loss (Loss) values are assessed for each run, and the results are plotted in Figure 5.

Figure 5 a) shows that the accuracies of all models first increase with increasing epochs and then stabilize. In the first 4 epochs, the nonneural models fluctuate considerably. During the stable phase, the accuracies of all models exceed 84 %, and the accuracy values of the two NN models are greater than 90 %. However, our model can reach 90 % accuracy with fewer epochs, so it has a good classification ability in a shorter training time.

As shown in Figure 5 b), the loss rate of our model is in the range of 0.2 % to 0.4 %, which is lower than that of the comparative models, and indicates that the convergence of the proposed model is better. The loss rate fluctuates once, which may be caused by sudden changes in some unknown factors, but it does not affect the overall trend. Overall, our model outperforms the comparative model in terms of convergence speed and accuracy.



a) Accuracy



b) Loss

Figure 5. Prediction distributions of the four models

6.2 User Study

We conducted user studies to evaluate the effectiveness of our visual analysis framework. Procedure and Participants: Three visualization experts and scholars were invited to discuss the evaluation metrics. Each expert had more than 5 years of experience with visualization. After discussion, the practicality, explanation,

effectiveness, readability and usability metrics were selected. Based on these five metrics and the analysis objectives of this paper, user studies were designed.

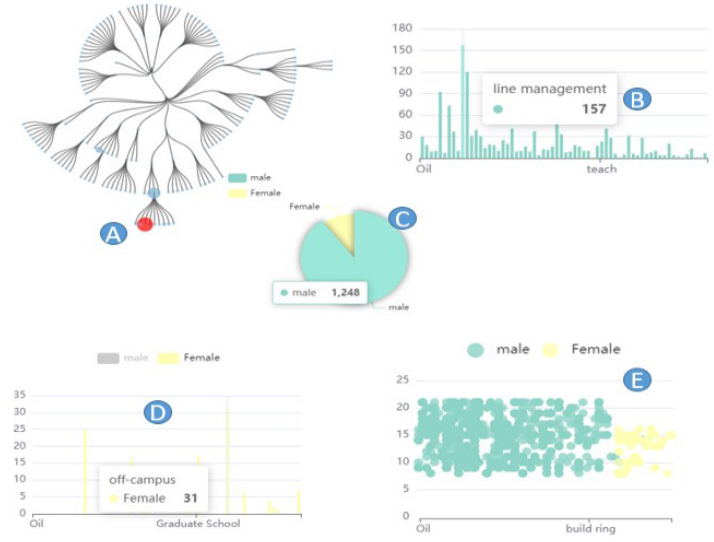
We conducted the study with ten volunteers from our school. The age range was 19 to 25 years, there were 7 males, 3 females, 6 undergraduates, and 4 graduates. Three of them had one year of experience in visualization, and the other had little knowledge of visualization. We selected a dataset that was familiar to the volunteers, namely, the book borrowing records dataset of a university library[1] and a literature books subset. First, we introduced the background, the data and the analysis task and then demonstrated the use of the mTreeIllustrator system. Subsequently, after a Q & A, the volunteers started their exploration. During their explorations, they were asked to record details they found meaningful or interesting.

Result and Analysis: Based on the exploration records, we interviewed the users; two representative use cases are shown in Figure 6 a) and Figure 6 b).

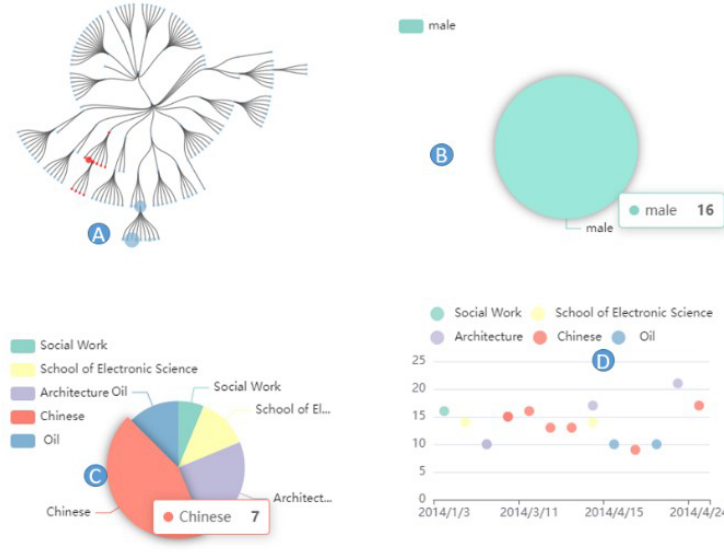
Volunteer 1's attention was first drawn to the hierarchical overview, where he found that the most popular books were romance novels, as shown in Figure 6 a) part A. The volunteer then wanted to know which majors contributed the most. He clicked on the node representing romance novels in the hierarchical overview, and the system automatically updated the overview view with only the romance novel data and generated top visualizations for the important attribute combinations in the multidimensional exploration view, from which the volunteer found the histogram of borrowing statistics for each major (B). Students in the "Administration" major contributed the most, followed by "Storage and Transportation" and "Financial Management". Then, he wanted to know the gender distribution, but the gender attribute was not selected by the recommendation pipeline. Therefore, he manually added that attribute, and the system regenerated top the charts according to the new selection. The volunteer first looked at the recommended pie chart (C) showing the percentages of male and female borrowers and found that the proportion of men was much larger than that of women. This phenomenon was unexpected; he thought that females would be the main readers of romance novels, but the proportion of males was much larger (D–E) in this dataset. The volunteer thought that was an interesting finding. Overall, the volunteer thought that the system could help users efficiently understand the characteristics of borrowing patterns.

Volunteer 2 focused on the prose branch on the hierarchical overview, as shown in Figure 6 b) part A, and added the gender attribute (B) to the currently explored attribute set. From the attribute exploration view, she found that the readers were mainly from "Architecture", "College of Electronic Science", and "Social Work" majors (C), and their borrowing dates were mainly March 2014 (D). The background of the readers shows that readers may be less interested in books in the prose category. Readers from the "Chinese" major contributed the most. This may be related to their course study needs.

[1] `https://github.com/wenbl/LibraryBigData/tree/master/data`



a) Volunteer 1



b) Volunteer 2

Figure 6. Exploration paths

The exploration results from the two volunteers illustrate that the automatic pipeline and the visual analysis system can help users quickly become familiar with mTree data and can also support efficient fine-grained exploration.

Usability Evaluation: After the users finished the experiment, they were asked to complete a questionnaire to evaluate the efficiency of the system. The assessment data are quantified using a five-point Likert scale, as shown in Figure 7.

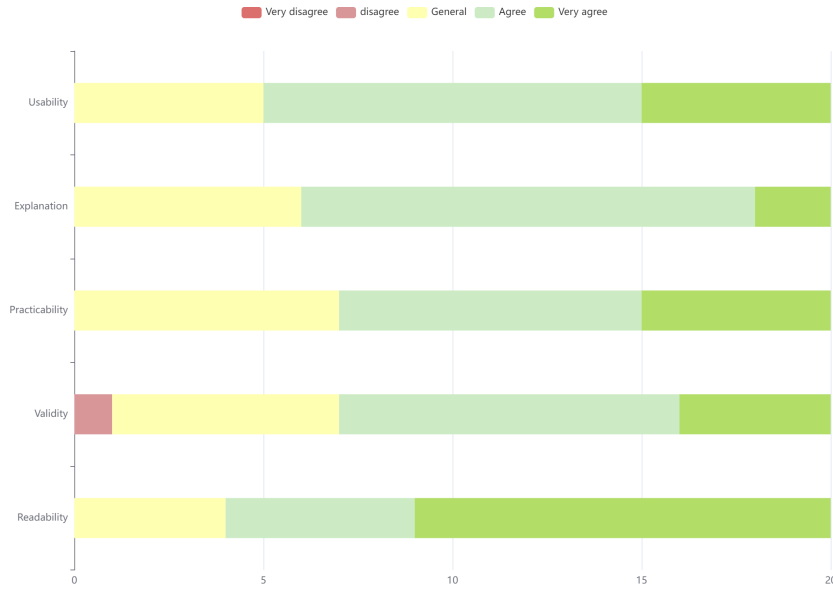


Figure 7. Score distribution of the questionnaire

Most volunteers agreed that mTreeIllustrator is useful, easy to learn, and easy to use. Each volunteer learned to use the tool quickly. When they were asked to compare their experience with that of previous library data exploration tools, they were all more in favor of this visual tool, saying the charts were easier to understand than the abstract data. Volunteer 3 said she especially liked the small charts in the left panel since they provided insight for further exploration. Among all metrics, the validity metric was slightly weaker than those of the other metrics. We interviewed the volunteers and found that the main reason for this score was that the attributes that the user wants to explore were occasionally not included in the recommendation list. For example, volunteer 1 manually added the gender attribute. This is a valuable finding; our model does not consider user differences, but in reality, people from different backgrounds do have different preferences. In the future, personalized learning algorithms would be studied.

7 CONCLUSION

In this paper, we presented mTreeIllustrator, a mixed-initiative framework for visual and interactive mTree data exploration. We proposed a machine learning-powered pipeline, consisting of an RF-based subspace importance evaluation model, a Bi-LSTM based visualization recommendation model and a rule-based chart encoding model, to automatically select the most important subspace from mTree data and encode the subspace into faceted visualizations. Moreover, we designed a visual framework and an interaction schema to couple the autogenerated visualizations with user selections to support progressive mTree data exploration. This approach also allows users to refine both the recommended visualizations and the data subspace, and to visually compare selected mTree structures. Comparative experiments and user studies demonstrated that our framework has good performance and can enable users to perform efficient and insightful mTree data exploration.

In the future, we plan to expand the range of our recommendation models. First, our model is purely data driven, and we plan to also consider the personal preferences and analysis goals to enable more diversified analyse. Another interesting area would involve studying the user interaction patterns exhibited during the mTree data exploration process and to developing models for providing navigation suggestions.

Acknowledgements

This research is jointly supported by the National Natural Science Foundation of China (No. 61872304) and the Talent Project of Sichuan University of Science and Engineering (No. 2020RC20).

REFERENCES

[1] Li, Y.: Hierarchical Data Visualization and Visual Comparison. Master Thesis. Shanghai Jiaotong University, Shanghai, 2016 (in Chinese).

[2] Chen, Y.—Zhen, Y. G.—Hu, H. Y.—Liang, J.—Ma, K. L.: Visualization Technique for Multi-Attribute in Hierarchical Structure. Journal of Software, Vol. 27, 2016, No. 5, pp. 1091–1102, doi: 10.13328/j.cnki.jos.004956 (in Chinese).

[3] Guerra-Gomez, J. A.—Pack, M. L.—Plaisant, C.—Shneiderman, B.: Visualizing Change over Time Using Dynamic Hierarchies: TreeVersity2 and the StemView. IEEE Transactions on Visualization and Computer Graphics, Vol. 19, 2013, No. 12, pp. 2566–2575, doi: 10.1109/TVCG.2013.231.

[4] Wongsuphasawat, K.—Qu, Z.—Moritz, D.—Chang, R.—Ouk, F.—Anand, A.—Mackinlay, J.—Howe, B.—Heer, J.: Voyager 2: Augmenting Visual Analysis with Partial View Specifications. Proceedings of the 2017 CHI Con-

ference on Human Factors in Computing Systems (CHI '17), 2017, pp. 2648–2659, doi: 10.1145/3025453.3025768.

[5] Inselberg, A.: Parallel Coordinates: Visual Multidimensional Geometry and Its Applications. Springer, New York, 2009, doi: 10.1007/978-0-387-68628-8.

[6] Schulz, H. J.: Treevis.net: A Tree Visualization Reference. IEEE Computer Graphics and Applications, Vol. 31, 2011, No. 6, pp. 11–15, doi: 10.1109/MCG.2011.103.

[7] Macquisten, A.—Smith, A. M.—Johansson Fernstad, S.: Evaluation of Hierarchical Visualization for Large and Small Hierarchies. 2020 24th International Conference Information Visualisation (IV), 2020, pp. 166–173, doi: 10.1109/IV51561.2020.00036.

[8] Li, G.—Tian, M.—Xu, Q.—McGuffin, M. J.—Yuan, X.: GoTree: A Grammar of Tree Visualizations. Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20), 2020, pp. 1–13, doi: 10.1145/3313831.3376297.

[9] Wongsuphasawat, K.—Moritz, D.—Anand, A.—Mackinlay, J.—Howe, B.—Heer, J.: Voyager: Exploratory Analysis via Faceted Browsing of Visualization Recommendations. IEEE Transactions on Visualization and Computer Graphics, Vol. 22, 2016, No. 1, pp. 649–658, doi: 10.1109/TVCG.2015.2467191.

[10] Moritz, D.—Wang, C.—Nelson, G. L.—Lin, H.—Smith, A. M.—Howe, B.—Heer, J.: Formalizing Visualization Design Knowledge as Constraints: Actionable and Extensible Models in Draco. IEEE Transactions on Visualization and Computer Graphics, Vol. 25, 2019, No. 1, pp. 438–448, doi: 10.1109/TVCG.2018.2865240.

[11] Qin, X.—Luo, Y.—Tang, N.—Li, G.: DeepEye: An Automatic Big Data Visualization Framework. Big Data Mining and Analytics, Vol. 1, 2018, No. 1, pp. 75–82, doi: 10.26599/BDMA.2018.9020007.

[12] Hu, K.—Bakker, M. A.—Li, S.—Kraska, T.—Hidalgo, C.: VizML: A Machine Learning Approach to Visualization Recommendation. Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19), 2019, doi: 10.1145/3290605.3300358.

[13] Pister, A.—Buono, P.—Fekete, J. D.—Plaisant, C.—Valdivia, P.: Integrating Prior Knowledge in Mixed-Initiative Social Network Clustering. IEEE Transactions on Visualization and Computer Graphics, Vol. 27, 2021, No. 2, pp. 1775–1785, doi: 10.1109/TVCG.2020.3030347.

[14] Horvitz, E.: Principles of Mixed-Initiative User Interfaces. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '99), 1999, pp. 159–166, doi: 10.1145/302979.303030.

[15] Liu, S.—Maljovec, D.—Wang, B.—Bremer, P. T.—Pascucci, V.: Visualizing High-Dimensional Data: Advances in the Past Decade. IEEE Transactions on Visualization and Computer Graphics, Vol. 23, 2017, No. 3, pp. 1249–1268, doi: 10.1109/TVCG.2016.2640960.

[16] Gomez, J. A. G.—Buck-Coleman, A.—Plaisant, C.—Shneiderman, B.: TreeVersity: Comparing Tree Structures by Topology and Node's Attributes Differences. 2011 IEEE Conference on Visual Analytics Science and Technology (VAST), 2011, pp. 275–276, doi: 10.1109/VAST.2011.6102471.

[17] Chen, Y.—Dong, Y.—Sun, Y.—Liang, J.: A Multi-Comparable Visual Analytic Approach for Complex Hierarchical Data. Journal of Visual Languages and Computing, Vol. 47, 2018, pp. 19–30, doi: 10.1016/j.jvlc.2018.02.003.

[18] Sakairi, T.—Ishida, A.—Achilles, H. D.: Visual Analysis Tool for Hierarchical Additive Time-Series Data. 2015 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), 2015, pp. 18–23, doi: 10.1109/SOLI.2015.7367610.

[19] Zhou, J.—Li, Z.—Zhang, Z.—Liang, B.—Chen, F.: Visual Analytics of Relations of Multi-Attributes in Big Infrastructure Data. 2016 Big Data Visual Analytics (BDVA), 2016, pp. 1–2, doi: 10.1109/BDVA.2016.7787052.

[20] Mackinlay, J.: Automating the Design of Graphical Presentations of Relational Information. ACM Transactions on Graphics, Vol. 5, 1986, No. 2, pp. 110–141, doi: 10.1145/22949.22950.

[21] Roth, S. F.—Kolojejchick, J.—Mattis, J.—Goldstein, J.: Interactive Graphic Design Using Automatic Presentation Knowledge. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '94), 1994, pp. 112–117, doi: 10.1145/191666.191719.

[22] Mackinlay, J.—Hanrahan, P.—Stolte, C.: Show Me: Automatic Presentation for Visual Analysis. IEEE Transactions on Visualization and Computer Graphics, Vol. 13, 2007, No. 6, pp. 1137–1144, doi: 10.1109/TVCG.2007.70594.

[23] Nan, M.—Xiaoru, Y.: Tabular Data Visualization Interactive Construction for Analysis Tasks. Journal of Computer-Aided Design and Computer Graphics, Vol. 32, 2020, No. 10, pp. 1628–1636 (in Chinese).

[24] Dibia, V.—Demiralp, C.: Data2Vis: Automatic Generation of Data Visualizations Using Sequence-to-Sequence Recurrent Neural Networks. IEEE Computer Graphics and Applications, Vol. 39, 2019, No. 5, pp. 33–46, doi: 10.1109/MCG.2019.2924636.

[25] Satyanarayan, A.—Moritz, D.—Wongsuphasawat, K.—Heer, J.: Vega-Lite: A Grammar of Interactive Graphics. IEEE Transactions on Visualization and Computer Graphics, Vol. 23, 2017, No. 1, pp. 341–350, doi: 10.1109/TVCG.2016.2599030.

[26] Cui, W.—Zhang, X.—Wang, Y.—Huang, H.—Chen, B.—Fang, L.—Zhang, H.—Lou, J. G.—Zhang, D.: Text-to-Viz: Automatic Generation of Infographics from Proportion-Related Natural Language Statements. IEEE Transactions on Visualization and Computer Graphics, Vol. 26, 2020, No. 1, pp. 906–916, doi: 10.1109/TVCG.2019.2934785.

[27] Breiman, L.: Random Forests. Machine Learning, Vol. 45, 2001, No. 1, pp. 5–32, doi: 10.1023/A:1010933404324.

[28] Battle, L.—Duan, P.—Miranda, Z.—Mukusheva, D.—Chang, R.—Stonebraker, M.: Beagle: Automated Extraction and Interpretation of Visualizations from the Web. Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18), 2018, doi: 10.1145/3173574.3174168.

[29] Saket, B.—Endert, A.—Demiralp, C.: Task-Based Effectiveness of Basic Visualizations. IEEE Transactions on Visualization and Computer Graphics, Vol. 25, 2019, No. 7, pp. 2505–2512, doi: 10.1109/TVCG.2018.2829750.

[30] Chen, L.—Yang, D.—Zhang, D.—Wang, C.—Li, J.—Nguyen, T. M. T.: Deep Mobile Traffic Forecast and Complementary Base Station Clustering for C-RAN Optimization. Journal of Network and Computer Applications, Vol. 121, 2018, pp. 59–69, doi: 10.1016/j.jnca.2018.07.015.

[31] Lee, C.—Kim, Y.—Jin, S.—Kim, D.—Maciejewski, R.—Ebert, D.—Ko, S.: A Visual Analytics System for Exploring, Monitoring, and Forecasting Road Traffic Congestion. IEEE Transactions on Visualization and Computer Graphics, Vol. 26, 2020, No. 11, pp. 3133–3146, doi: 10.1109/TVCG.2019.2922597.

[32] Shneiderman, B.: The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. Proceedings 1996 IEEE Symposium on Visual Languages, 1996, pp. 336–343, doi: 10.1109/VL.1996.545307.

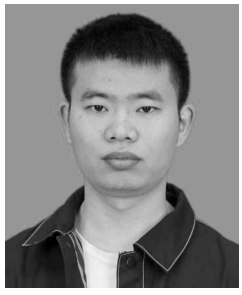
[33] Zhou, M.—Tao, W.—Pengxin, J.—Shi, H.—Dongmei, Z.: Table2Analysis: Modeling and Recommendation of Common Analysis Patterns for Multi-Dimensional Data. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 320–328, doi: 10.1609/aaai.v34i01.5366.

[34] Miller, G. A.: The Magical Number Seven Plus or Minus Two: Some Limits on Our Capacity for Processing Information. Psychological Review, Vol. 101, 1956, No. 2, pp. 343–352, doi: 10.1037/0033-295X.101.2.343.

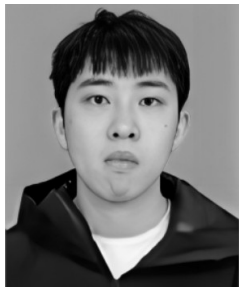
[35] Wei, X.: Analytical Deep Learning: Convolution Neural Network Principles and Visual Practice. Electronic Industry Press, 2018 (in Chinese).

Guijuan Wang received the M.Sc. degree from Beihang University, Beijing, China in 2007. She is currently pursuing the Ph.D. degree in the School of Information and Technology of the Southwest University of Science and Technology. Her research interests include intelligent city visualization and automatic visualization.

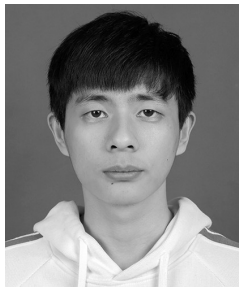
Yu Zhao received his Ph.D. degree from the Brunel University London, UK in 2014. His research interests include cloud computing, data analytics and artificial intelligence. He is currently Assistant Research Scientist at the Shandong Academy of Social Sciences.



Boyou Tan received his M.Sc. degree from the Southwest University of Science and Technology, Mianyang, China, in 2022. His major research interests include information visualization and mobility pattern.



Zhong Wang received his B.Sc. degree from the Southwest University of Science and Technology, Mianyang, China, in 2019. He is currently pursuing his M.Sc. degree in the School of Computer Science and Technology of the same university. His research interests include information visualization and mobility pattern.



Jiansong Wang received his B.Eng. degrees in computer science and technology from the Southeast University, Chengxian College in 2020. He is currently Master student at the Southwest University of Science and Technology. His research interests are data visualization and digital twin.

Hao Guo is M.Sc. student at the Southwest University of science and technology. He is interested in data visualization and natural language processing.

Yadong Wu is Dean at the School of Computer Science and Engineering, Sichuan University of Science and Engineering, Zigong China. He finished his Ph.D. at the University of Electronic Science and Technology of China. His current research interests include visualization, virtual reality and digital twins technology.

Computing and Informatics, Vol. 42, 2023, 716–740, doi: 10.31577/cai_2023_3_716

INTEGRATION OF A CONTEXTUAL OBSERVATION SYSTEM IN A MULTI-PROCESS ARCHITECTURE FOR AUTONOMOUS VEHICLES

Ahmed-Chawki Chaouche

MISC Laboratory, University of Constantine 2 – Abdelhamid Mehri Ali Mendjeli Campus, 25000 Constantine, Algeria
e-mail: ahmed.chaouche@univ-constantine2.dz

Jean-Michel Ilié

LIP6, UMR 7606 UPMC – CNRS 4 Place Jussieu 75005 Paris, France
e-mail: jean-michel.ilie@lip6.fr

Assem Hebik

MISC Laboratory, University of Constantine 2 – Abdelhamid Mehri Ali Mendjeli Campus, 25000 Constantine, Algeria
e-mail: assem.hebik@univ-constantine2.dz

François Pêcheux

LIP6, UMR 7606 UPMC – CNRS 4 Place Jussieu 75005 Paris, France
e-mail: francois.pecheux@lip6.fr

Abstract. We propose a software layered architecture for autonomous vehicles whose efficiency is driven by pull-based acquisition of sensor data. This multi-process software architecture, to be embedded into the control loop of these vehicles,

includes a Belief-Desire-Intention agent that can consistently assist the achievement of intentions. Since driving on roads implies huge dynamic considerations, we tackle both reactivity and context awareness considerations on the execution loop of the vehicle. While the proposed architecture gradually offers 4 levels of reactivity, from arch-reflex to the deep modification of the previously built execution plan, the observation module concurrently exploits noise filtering and introduces frequency control to allow symbolic feature extraction while both fuzzy and first order logic management are used to enforce consistency and certainty over the context information properties. The presented use-case, the daily delivery of a network of pharmacy offices by an autonomous vehicle taking into account contextual (spatio-temporal) traffic features, shows the efficiency and the modularity of the architecture, as well as the scalability of the reaction levels.

Keywords: Autonomous vehicle, multi-process architecture, context-awareness, contextual planning, reactive behavioral strategies, logical context modeling

1 INTRODUCTION

The design of autonomous vehicles is a highly active area of research. Developing a vehicle which is able to observe, plan and react safely with the surrounding environment is a major challenge for both researchers and industrialists [1].

Different works in the autonomous vehicles domain and in particular on robotics enhance some cognitive architectures dedicated to the representation of the human mind. In particular, the symbolic architecture SOAR [2] is built on a two layered system to capture both the human cognition processes and the operational activities. Related concepts can also be modeled, such as attention and the motivations which can have an impact on the design of an intelligent system [3, 4]. These works mostly suggest a system composed of many connected processes, each one representing a specific sub-task [5].

Moreover, various agent models have already been proposed to handle the ambient context. In particular, the Belief-Desire-Intention (BDI) approach which has the advantage of introducing smart software agents with high level reasoning capacity, mainly in terms of intentions (I), coming from agent Beliefs (B) and Desires (D) [6]. Since these native basic agents lack context awareness capacities, authors of [7] proposed a reactive model as a supplement to the agent APL programming language in order to control the software components of a robot. This work is related to software multi-layered architectures, like 3T, ATLANTIS and LAAS [8], which all subsum the agent behavioral information with the price to handle all the event messages at the deliberative/planning layer. In this sense, the standard ROS operating system [9, 10] emerges greatly in order to simplify the implementation of operational physical robotic architectures.

Our main objective is to provide the autonomous vehicle with a guidance mechanism that computes an execution plan of actions while taking into account unex-

pected changes in the context. Thus, the main issue comes from the various and numerous asynchronous events that may occur in the ambient environment that can jeopardize the resilience of the vehicle. Being context aware, this vehicle needs to identify its context correctly (traffic, road signs and traffic light, obstacles, current location, etc.) to make the suitable decision at the right time. The context information is acquired using sensors which may be physical (from hardware source), virtual (from software applications or services) or logical (from composition of several sources) [11] and passed as raw data inside a context aware system in order to be analyzed, filtered and symbolized.

Due to its complexity, context-awareness is the subject of many works and studies. In [12], context aware system architectures are introduced and deployed in different applications. Other studies about context modeling and reasoning techniques were presented in [11, 13, 14]. These works explore existing context aware systems, context modeling approaches and identify the major challenges and requirements for such systems.

Moreover, various techniques for data acquisition and preprocessing are presented in [15]. These studies present the different existing models for acquiring sensor data, in addition to data smoothing and noise filtering techniques. Generally, the design of a context aware system can be divided into sub-problems, consisting first of acquiring the context, then modeling and reasoning about it. In particular, the fact that sensors are relatively uncertain sources of information, requires the model to be able to handle uncertainty while being consistent and expressive. Another important aspect is the ability to represent relations between context information, which helps in the reasoning phase to deduce more information about the context and to check its consistency.

In this paper, we opt for Embedded Higher order Agent (E-HoA) architecture [16], dedicated to context-aware autonomous vehicles. This architecture embeds the high level BDI agent HoA [17] in a ROS-based platform. Actually, HoA agents are particularly well-suited to handle the concurrency of intentions and learn from past contextual information to provide appropriate execution plans that will be successfully achieved if applied in a new but yet similar context. The presented work can be viewed as an extension of the HoA approach to help the decision making of a concrete self-driving vehicle. The E-HoA architecture has two main advantages:

1. From the ROS viewpoint, the agent is context-aware, it can learn from field information and can react in real-time;
2. From the E-HoA viewpoint, reasoning on context information helps generate symbolic intentions concretized as a plan of actions that can be scheduled then performed at ROS level.

For the sake of efficiency, we aim to develop E-HoA architecture like a new distributed platform based on multi-processes and a client-server protocol allowing both synchronous and asynchronous communications. This is used to decentralize the agent decision center in several pieces, while facilitating a coherent context-

awareness through subscriptions to services managing context information. Also, we aim at showing that we can benefit from this decentralization to graduate the vehicle reaction at different levels.

To reason on context efficiently, we propose a contextual observation system, which offers context modeling and reasoning mechanisms occurring between sensors and *E-HoA* layers. At ROS level, a specific node, called *Acquire*, is responsible for collecting context raw data from different sensors, controlling the frame rate then denoising and smoothing the raw data. The resulting data are delivered to an *Observation process* situated at the E-HoA level. Next, the processed data are passed to a helpful fuzzy logic processor in order to be symbolized, knowing that sensors can provide misleading values sometimes and that the vehicle cannot have a universal knowledge of the real context, therefore inconsistency problems are very likely to happen. To solve such problems, we use an expressive context model that takes into account the certainty of context information, and helps in the reasoning process to detect inconsistencies. Furthermore, the reasoning process allows us to infer new context information based on the specification of defined rules and relations.

The outline of the paper is as follows: Section 2 presents the multi-process E-HoA architecture, which is able to execute the vehicle intentions and actions on a ROS system, by means of contextual planning and learning mechanisms. The nominal loop of the vehicle behavior is detailed. Section 3 identifies and details four different levels of reactions, trying to maintain much of the vehicle intentions. In Section 4, we present how context observation is achieved following the life cycle of sensed data from acquisition to reasoning followed by a brief sample demonstrating the utility of our approach. In Section 5 for efficiency purposes, we show how to deploy the E-HoA processes on a concrete distributed platform. Then, a delivery use case is presented based on a city road map to demonstrate the E-HoA interest in practise. Section 6 discusses our approach with regard to related works. The last section concludes and outlines our perspectives.

2 E-HOA LAYERED ARCHITECTURE

Like many autonomous driving systems, the goal of the proposed E-HoA architecture is to develop the fixed computational building blocks necessary for general cognitive agents. Those agents can perform a wide range of tasks like path planning, decision making, or problem solving. E-HoA is a computational implementation of a theory that combines BDI reasoning concepts and their physical concretization as a set of maneuvers for an autonomous vehicle, while at the same time considering the dynamic evolution of its surrounding spatio-temporal context.

As stated by Figure 1, E-HoA architecture is composed of four layers that are vertically tightly coupled to achieve a good level of performance and accuracy. Concretely, E-HoA consists of a set of cooperating processes that altogether define the robot behavior by exchanging synchronous and asynchronous messages using a publish/subscribe paradigm and taking advantage of a graph-based database for plan-

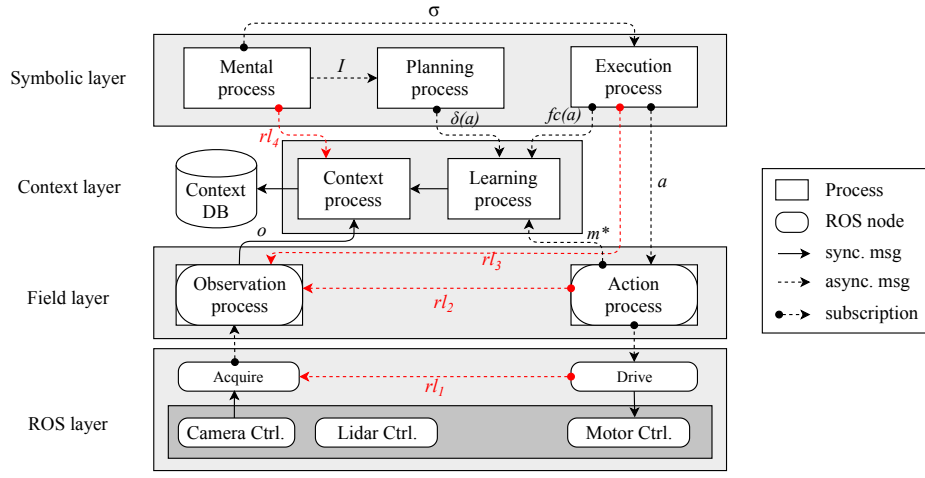


Figure 1. Embedded Higher-order Agent (E-HoA) architecture

ning. The lower layer instantiates two ROS nodes (*Acquire* and *Drive*) to allow E-HoA to be interfaced with all the available ROS building blocks such as sensor/actuator libraries or higher level software components such as Simultaneous Localization And Mapping (SLAM) proposed by the vivid ROS community [10]. The context layer is of particular interest as it constitutes the long and short term memory needed at all the levels of learning and reasoning.

The Symbolic Layer. All the high-level decisions of the E-HoA agent are taken at the symbolic layer according to its context information. The major process in this layer is the *Mental process* which reasons in terms of Beliefs (B), Desires (D) and Intentions (I) [6]. Aiming at optimizing the achievement of the agent's intentions, the Mental process asks the *Planning process* on the same layer to compute an optimal plan of symbolic actions (σ), with respect to the original intentions (I) and the available context information data. Then, the Mental process asks the *Execution process* to perform in order, the actions defined by the plan.

The Field Layer. The field layer is the concrete layer of the E-HoA architecture. In practice, the *Action process* of the field layer receives symbolic actions from the Execution process and converts each symbolic action (a) into a finite set of implemented maneuvers ($m*$) controlling the robot operations. To provide context-awareness, a second process called *Observation process* is responsible for capturing the real-world physical values from the robot and its ambient environment that will be abstracted and symbolized (o) to enrich the context layer. The Observation process mainly aims at acquiring raw or abstracted information from the different sensors and actuators.

The ROS Layer. The Action and Observation processes of the E-HoA architecture are in direct contact with their ROS nodes counterparts that manage the sensors (LIDAR, camera, IMU) and actuators (left and right motors). This layer relies on ROS and simplifies the interfacing of E-HoA with real robotic systems. In particular, it allows the seamless shift from a simulated vehicle and environment modeled with Gazebo (ROS modeling and simulation tool) to a physical robot operating in a real world.

The Context Layer. The context layer is inserted between the symbolic (Higher layer) and field (Middle layer) layers. It is composed of two main processes. The first one, namely the *Context process*, aims at storing the observed context information for later retrieval. Like the Observation process, it also acts as an information provider other processes can subscribe to. Three kinds of symbolic information managed in practice: The *state context* manages the state of the robot elements and also the environmental information (weather consideration, states of the road map and of the different environmental objects); the *execution context* yields the current state of the execution plan; finally, the *historical context* contains information about the performances of the robot activities, in terms of intentions, plans, actions and maneuvers. The second process of this layer is the *Learning process*. This pivotal process learns about the context information in order to help decide some optimization criteria [18]. For instance, it optimally computes the best path between several locations, by managing a road map viewed as a graph and estimating the transit durations of the road map sections.

2.1 Inter-Processes Communication

Altogether, the three upper layers (symbolic, context and field) cooperate and exchange information to consolidate behavior of the robot at all times. E-HoA architecture as a distributed system is a set of concurrent processes with coordinate and communicate thanks to services according to a client/server (synchronous) approach or a publisher/subscriber (asynchronous) formalism. Messages exchanged fall into one of the three following categories:

Synchronous message which provides a simple transmission scheme: Client process sends a request and waits for an immediate reply message from the requested server.

Asynchronous message which provides a bidirectional scheme: Client process sends a request and is notified by the server with one immediate or delayed reply message.

Subscription message which provides a publish/subscribe mechanism, thus extending the asynchronous message scheme: Client process sends a subscription request to be notified with several intermediate responses coming from the server process. Such a subscription scheme is useful when a client needs milestone re-

porting about the progress of a significantly long operation such as the conversion of a symbolic action into the corresponding set of maneuvers.

2.2 E-HoA Execution Loop

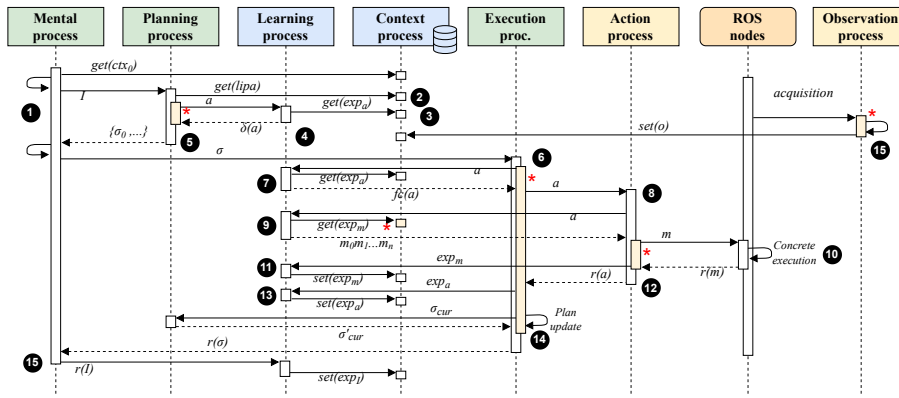


Figure 2. Nominal E-HoA execution loop

The nominal E-HoA loop addresses first the execution of an intention by means of some successive refinements up to the concrete execution of maneuvers. This principle is highlighted here due to the fact that a set of intentions are executed concurrently.

Figure 2 is an UML sequence diagram that details how the four layers of the E-HoA architecture cooperate to define its behavior. It all begins with a starting set of intentions I acquired by the Mental process (1). The mental process has to compute an execution plan (a set of ordered symbolic actions) and is assisted in its task by the Planning process which analyses the different plans associated with the intentions according to an available spatio-temporal context. The Planning process can eventually get information from a library of action plans (2) available through the context layer. For each action a of the actions related to an intention I, the Planning process may ask the Learning process some experience data $get(exp_a)$ (3) to evaluate the duration of a (4). The Planning process may then accumulate all the duration-weighted actions to return a list of feasible sequences of actions $\{\sigma_0, \dots\}$, among with an optimal one (σ) in terms of duration $\{\delta(a)\}$ (5). Thus, Planning is a complex process as it may require the service of the Learning process to get a good estimation of the duration of each considered action a concretely (for instance, see [18]). It also should be noticed that the ROS node responsible for data acquisition and dynamically notifies the Observation process with environmental data that, once correctly abstracted, are used to feed the context (15).

In general and with respect to the currently available context, the Mental process selects one optimal action sequence σ and then delegates its achievement to the

Execution process (6), which is responsible for the correct overall execution of the sequence. It is helped by the Learning process (7) which can determine the potential failure conditions restraining the execution of the actions ($fc(a)$). In order to control the concrete execution of actions, a maximum timeout duration is computed for each action and is considered the single condition which triggers the failure of the action. The Execution process can then delegate the concrete execution of action a to the Action process (8).

For all the consolidated actions, the Action process asks the Learning process (hence also the Context process) the list of learned corresponding maneuvers (9) and then performs them in order. To actually compute the most efficient decomposition, the Action process may ask the Learning process for the Contextual Shortest Path (CSP) to a given point on the map (according to the evolution of the spatial-temporal context).

Hence, each symbolic action is decomposed into a series of individual maneuvers that are propagated to the vehicle motors. The Action process is directly connected to the ROS node that actually drives the vehicle and materialize the execution (10).

Once a specific maneuver is completed, it notifies the Action process with the result of m (denoted $r(m)$), success or failure (11). The success or failure of a specific maneuver reinforces the E-HoA experience (exp_m) and the context database is updated accordingly (11).

When the list of maneuvers corresponding to an action a has been entirely processed or in contrast when a problem is detected from some maneuver, the Action process notifies the corresponding outcome of a ($r(a)$) to the Execution process (12). As before, the success or the failure of an action a may be used to increase the E-HoA experience (exp_a). The context database is updated accordingly (13). The Execution process can then proceed to the update of the execution plan with respect to the actual duration time for action a, and thus accumulate experience on its concrete realization (14).

When an execution plan composed of a list of actions has been fully completed or in contrast when one action turns in failure, the Mental process is notified of the intentions that are achieved or failed (15). The Mental process can then deliberate implying possible changes in the considered set of the intentions, before retriggering the so-called nominal loop. It is worth noticing that the intentions that remains in activity can simply be resumed from their reached execution state, as in [19].

3 MULTI-LAYERED AND CONTEXT REACTIVE STRATEGIES

The E-HoA layered architecture provides four means to handle external or unexpected events, each of which depending on the complexity of the appropriate handling routine to be executed. These handling routines correspond to four reactivity levels (rl_i), depicted as red connection lines in Figure 1. E-HoA is thus able of adapting itself to evolution and changes of the spatio-temporal context. From a system viewpoint, unexpected events correspond to interrupts with respect to the previously

described nominal execution loop. Accordingly, the four reactive levels correspond to the four levels of Interrupt Service Routines (ISR) provided by E-HoA.

Figures 3, 4, 5 and 6 are UML sequence diagrams that respectively detail the four reactive levels noted rl_1 to rl_4, according to the duration and latency of their management (from the simplest and quickest rl_1 that involves only the ROS layer to the most complex rl_4 that may impact the whole architecture).

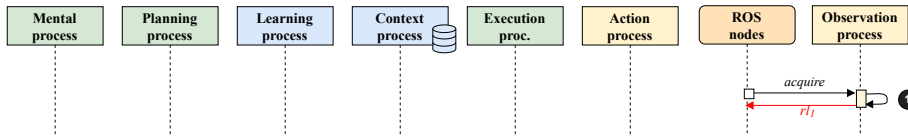


Figure 3. Functioning of arch-reflex (rl_1) strategy

The lowest reactivity level, rl_1 or arch-reflex, operates only at the ROS layer level, and represents the ability of E-HoA agent to have vehicle reflex capabilities, i.e. the ability to react with a very small latency to immediate events that would, if not correctly and quickly handled, cause trouble to the vehicle (car crash) or the environment (person injury when the vehicle runs into a human being). The different sensors on the vehicle (LIDAR, distance sensors) and the two ROS nodes (*Acquire* and *Drive*) cooperate to constitute altogether a pre-mitigation braking system that can avoid or get around obstacles (1). From an architectural viewpoint, the ROS action node subscribes to the observation topics serviced by the ROS Acquire node (hence the direction of the arrow in the rl_1 connection). In practice, the appropriate response is to successively stop the currently executed maneuver, execute the *"get around"* maneuver service routine, and resume the executed maneuver. It is worth notifying that the upper layers could not be notified with this local modification of maneuvers.

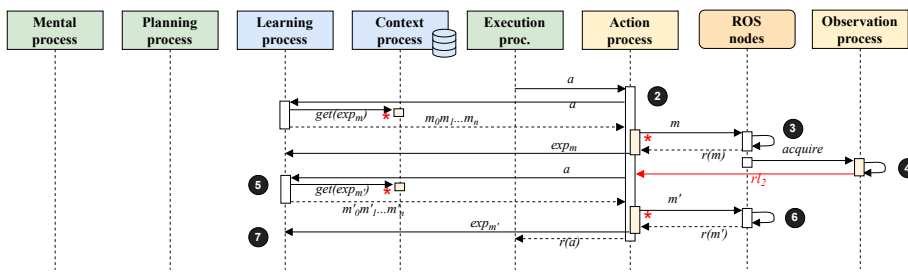


Figure 4. Functioning of field-reflex (rl_2) strategy

The second reactivity level, rl_2 or field-reflex, allows the E-HoA agent to correctly handle situations where a specific maneuver m cannot be achieved, due to an unexpected spatio-temporal condition (a specific section of the path to be followed by the vehicle as part of its maneuvers corresponding to the current execution plan

happens to be an unexpected traffic jam). Figure 4 shows how the management of such a case is dealt with by E-HoA. Consider an action issued by the Execution process and sent to the Action process, which in turn asks the Learning process to give back the correct sequence of maneuvers to be performed (2). Once the sequence is obtained, the corresponding list of maneuvers is executed in order (3). The ROS Acquire node may notify the Observation process with the event of an intractable maneuver (4), corresponding to an rl_2 reflex. In that case, the Action process has to request from the Learning process an alternative action, with its associated maneuvers m'_0 to m'_n (5). The calculated sequence of maneuvers is then executed as a whole (6), provided no blocking event is detected during this re-execution (7).

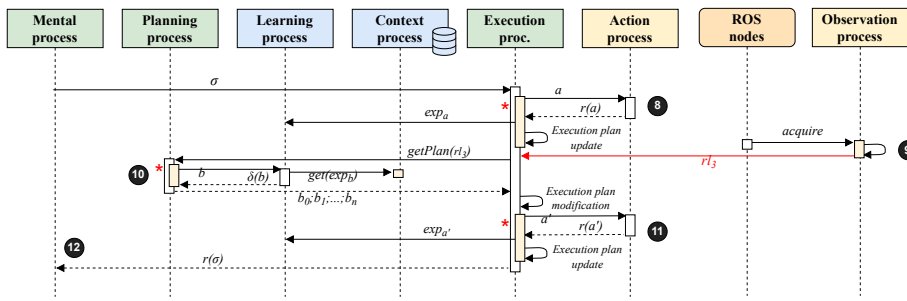


Figure 5. Functioning of action-reflex (rl_3) strategy

The third reactivity level, rl_3 or action-reflex, is activated when a specific action, part of an execution plan cannot be achieved anymore, due to the accumulated delays resulting from the execution of the previous actions in the execution plan or due to specific and urgent conditions such a *"battery low"* event coming from the Observation process. On the reaction diagram of Figure 5, this action-reflex occurs during the nominal execution of an action (8). When this event occurs, originating from the ROS Acquire node and Observation process, it is directly sent to the Execution process, that has to take the appropriate steps to modify the current execution plan, according to the new context (10). This involves recomputing the new execution plan σ', including the new actions b_0 to b_n (11). Once calculated, the updated execution plan is communicated to the Execution process that obtains a new chance to perform it until its successful completion (12).

The fourth reactivity level, rl_4 or intent-reflex, impacts the whole E-HoA architecture because it has a direct effect on the symbolic layer and the intentions considered by E-HoA agent. Events that can lead to the global re-evaluation of intentions can be related to global environmental conditions such as weather changes. Considering a currently set of intentions I (13), if the sensors detect a major change in a condition (snowfall in several but not all regions) that can sensibly modify the correct achievement of the intention (14), the Mental process must be notified with this global condition *"snowfall forecast"*, that is simultaneously saved in the context database. Once warned, the Mental process deliberates and possibly discards the

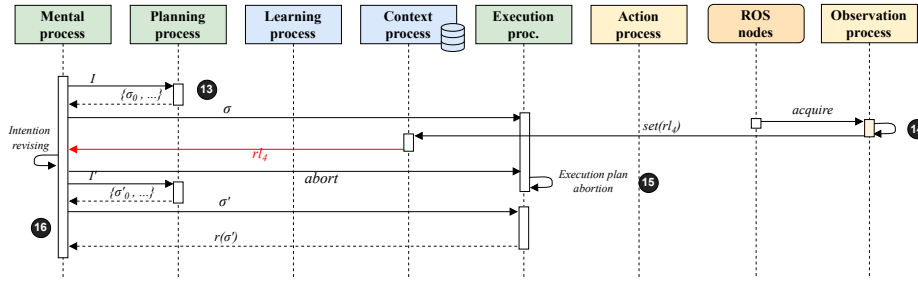


Figure 6. Functioning of intent-reflex (rl_4) strategy

intention (15) and a new planning-execution schema is generated that takes this global contextual change into account (16).

4 CONTEXTUAL OBSERVATION SYSTEM

Context-aware systems are responsible for raw data acquisition from sensors, noise reduction, and data-clearing. The acquired data passed then into features extraction. This low level data is used in the reasoning process for aggregation and composition then for validating the consistency of these acquired data.

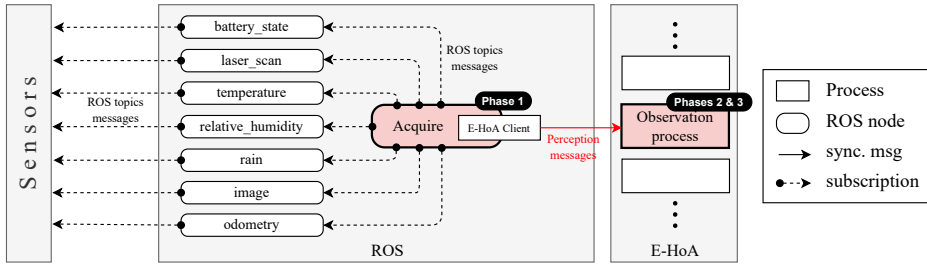


Figure 7. Observation System Architecture

The proposed architecture in Figure 7 describes a distributed context aware system which uses a blackboard model to serve context data acquired by sensors. Components of the system use neither the same protocols nor the same type of messages, however, each two components directly connected must use the same protocol to be able to send and receive messages between each other.

The proposed contextual observation system is organized in three segments: *Sensors*, *ROS* and *E-HoA*. This allows us to divide the complex tasks into smaller and simpler ones while adding more flexibility to the system. Moreover, a whole segment may be distributed or replaced without any breaking changes to the other segments.

The Sensors segment represents sensors of all types, having one basic goal which is sending raw data from sensors to the ROS segment. Sensors may use any protocol and any type of message when sending their data as long as the adapter node in the ROS segment implements the same protocol.

The ROS segment is a software part that acts like a bridge between Sensors and E-HoA layers. In particular, Acquire node is responsible for frequency control and noise-reduction of sensed raw data: It reads data from sensor adapters using ROS topics in subscriber mode; Once the data is acquired, it controls the frequency and reduces noise; Then, it analyzes processed data and outputs perception messages.
Acquire node implements an E-HoA client in order to send synchronously perception messages to the Observation process of E-HoA.

The E-HoA segment includes all the E-HoA processes, in particular Observation one. After receiving filtered context data from Acquire node, the Observation process applies low level and high level processings: First, it symbolizes the filtered context data with a fuzzy logic mechanism, then the resulting data can be processed with a first order logic technique which applies composition or aggregation to them.

In order to obtain relevant context data, we proceed in three successive stages: The acquisition of context data, its symbolization and reasoning about.

4.1 Data Acquisition Phase

In this phase, the Acquire node applies noise reduction and frequency control of sensed raw data. The fact that the sensors are not precise even if the quality of the sensor is high, a noise reduction mechanism is necessary to eliminate the inaccurate values. Noise reduction also called data clearing may be achieved using many different models. Regression models are considered among the most efficient models, these models compute the dependency from sensor value with respect to time, and then consider the regression curves as standards over which the sensor values reside. Otherwise, probabilistic models can also be used for data clearing. The expected natural range for the next sensed value is calculated based on the previous values. As in [15], the value is then excluded if it is outside the calculated field.

In our approach, we use the Savitzky-Golay smoothing filter [20] which performs a local polynomial regression of degree K on a series of sensed values to determine for each one the smoothed value. It thus preserves distribution features of values such as relative maxima, minima and width.

Let $\mathcal{D}$ be the set of all possible sensed values. Applied in a period of time Δt, we use the Savitzky-Golay filtering function $\mathit{filter} : 2^{\mathcal{D}} \times \mathbb{N} \times \mathbb{N} \to 2^{\mathcal{D}}$. The applying of $\mathit{filter}(D, \mathit{win}, \mathit{deg})$ corresponds to filtering the set D of input raw values, such that win is the number of values to consider when smoothing each value, and deg is the order of the polynomial that will be fitted to those raw data ($\mathit{deg} < \mathit{win}$).

For example, to filter the raw data from temperature sensor, we can apply $F_{temp} = filter(D_{temp}, 5, 2)$.

In addition to noise, the frequency of sensing can vary according to the physical sensor types. In the case of very high frequency sensors, the observation system can be quickly overwhelmed with a massive amount of data. Although we can decrease this frequency and save energy by configuring sensors, we can take advantage of high frequencies to increase the accuracy of information. One solution would be to collect the filtered values in a container for a specific period, then calculate only one value that represents all the other ones. We use the frequency control function $freq : 2^{\mathcal{D}} \rightarrow \mathcal{D}$. For instance, we can simply use the average function $freq(F) = \sum_{v \in F} v/|F|$, where F is the set of filtered raw data.

Indeed, this solution should not be overused at the expense of other properties such as excessive energy consumption or over-exploitation of computational resources.

Once filtered and smoothed, the sensed data is wrapped by the Acquire node in a standard message, called the *Perception message*. In addition, the Perception message contains the information of the sensor which acquired the data as well as the moment of acquisition. Thus, the Acquire node swallows different types of messages (via ROS topics) into a one standard perception message, which has the following global shape:

```
PerceptionMessage = {
   "type"   : "temperature",
   "params" : { "value" :  21, "unit" : "C", "seq" : 101,
      "timestamp" : 1594672461 }
}
```

4.2 Symbolization Phase

The goal of context symbolization is to transform raw data into atoms, called *context information*. Upon receiving a perception message from Acquire node, the Observation process performs the extraction of context information. There are many techniques that can be used to give the received raw data specific meanings. Each of these techniques has a particular goal in a particular use case as stated by [12]: Some techniques have been used for detection, classification and identification, like neural networks or Bayesian networks. Other techniques have been proposed to deal with uncertainties of data, such as logical templates, knowledge bases and fuzzy logic.

In our symbolization approach, we opt for fuzzy logic for its expressiveness and flexibility. The fuzzy logic can handle problems with imprecise and incomplete data. Further, at the symbolization level, it should not have a strict judgment on the context information, which helps the Observation process to reevaluate these judgments in the reasoning phase to be consistent with the other context information.

Definition 1 (Context information)**.** The context information is a tuple $\langle element, value, certainty \rangle$, where *element* is the context element to be described, *value* is either a value/state of this element, and $certainty \in [0, 1]$ is the extent of the credibility of the information.

The Observation process receives a sensor acquired value as an input and produces a context information as an output. Processing a sensor value without considering other sensor values around its context could lead to undesirable results. Therefore, all relevant values from other sensors are merged together to obtain more precise and relevant context information. For instance, the weather context information does not only depend on the temperature value, other factors such as relative humidity, altitude, season, and day period also matter.

In the following example, weather context information is produced from only temperature and relative humidity. The possible labels for these variables are:

- $temperature \in \{low, average, high\}$,
- $humidity \in \{low, average, high\}$,
- $weather \in \{cold, normal, hot\}$.

Many fuzzy membership functions exist, the ones which can be applied for temperature variables are: linear function for *low* and *high*, and triangular function for *average*. The set of rules for this example is:

- if temperature is *average* and humidity is *average* then weather is *normal*,
- if temperature is *low* and humidity is *high* then weather is *cold*,
- if temperature is *high* and humidity is *low* then weather is *hot*.

For example, if the temperature is 26°C and humidity 43 % then the resulting weather context information is $\langle weather, cold, 0.1 \rangle$ and $\langle weather, normal, 0.7 \rangle$ and $\langle weather, hot, 0.2 \rangle$.

As with weather context, all other types of context can be symbolized in the same way but using different variables and rules. However, object detection in image raw data is achieved using the YOLOv3 algorithm [21], as it is one of the best algorithms for applying real time detection accurately. The YOLOv3 classification process produces also context information with the same parameters, like $\langle TrafficLight, red, 0.75 \rangle$ and $\langle TrafficLight, orange, 0.25 \rangle$.

4.3 Reasoning Phase

In the reasoning phase, we first represent context information in a consistent form using first order logic predicates by using Prolog language, then we apply composition and aggregation of context information and a consistency validation. For context modeling, we need to represent information in a generic, consistent and expressive model which makes reasoning more efficient and easy to do.

In this paper, the context information are written in logic expressions by using SWI-Prolog tool [22]. In fact, SWI-Prolog is the most popular implementation of Prolog and supports a large number of features.

In this phase, we consider two types of context information: *knowledge* or facts that have certainty equaling to 1, and *assumptions* that have certainty between 0 and 1. In Prolog, the context information is expressed by the predicate $ctx(element, value, certainty)$. For the sake of clarity, the knowledge predicate $ctx(element, value, 1)$ is simply expressed as $ctx(element, value)$. For example, the assertion *"It is hot in Paris at night"*, can be expressed in SWI-Prolog as:

```
ctx(temperature, hot) * ctx(place, 'Paris') * ctx(period,
   'night').
```

In order to group many context information having the same context, we use the binary operator "*", instead of operator "and", to make the final expression unbreakable. This operator already exists in SWI-Prolog, however we simply define it in any other logical language.

Relations between context information is a critical factor in reasoning, thus a good representation of relations makes reasoning more efficient. Relations are expressed with only one predicate, like for context information.

Definition 2 (Context relation)**.** A context relation is a binary relation represented by a tuple $\langle term_1, relation, term_2, correlation\rangle$, where $term_1$ and $term_2$ are the related operands of the relation, *relation* is the name of relation and $correlation \in [0, 1]$ is the relation value for some relations that needs to be estimated.

In Prolog, the context relation is simply expressed by the predicate $rel(term_1, relation, term_2, correlation)$.

Relations are defined by their properties before they are used. We use the predicate $define(relation, property)$ to define the logical property relation *relation* with the binary relation property *property*. All the possible relation properties are supported. Using SWI-Prolog, we can trivially define the axioms for the most used properties (reflexive, transitive and antisymmetric):

```
rel(A, R, A, V) :- define(R, reflexive).
rel(A, R, B, V) :- define(R, symmetric), break(B, R, A, V).
rel(A, R, B, V) :- define(R, transitive), break(A, R, I, V),
   rel(I, R, B, V).
```

In order to avoid infinite loops in axiom definitions, we use break predicate to break it. Like for context information predicate, we simply express the relation with absolute correlation $rel(term_1, relation, term_2, 1)$ as $rel(term_1, relation, term_2)$.

4.4 An Illustrative Sample

Specifically, to apply the reasoning phase, we need to define each relation by its properties, then we specify rules and facts. Finally we query the knowledge base for results.

Rules Definition: In the following sample, we use three relations *locatedIn*, *near* and *surroundedWith*.

```
define(locatedIn, transitive)
define(near, symmetric)
```

Rule Set: We add two rules to demonstrate the efficiency of reasoning phase and the simplicity of rules definition:

Rule 1: If $PLACE_A$ is located in $PLACE_B$ and temperature is $TEMP$ in $PLACE_B$ with certainty of $CERT$, then the temperature is $TEMP$ in $PLACE_A$ but with lower certainty say like $0.9 * CERT$.

```
ctx(temperature, TEMP, NEW_CERT) * ctx(place, PLACE_A,
   1) :-
rel(PLACE_A, locatedIn, PLACE_B,1),
(ctx(temperature, TEMP, CERT) * ctx(place, PLACE_B, 1)),
NEW_CERT is (0.9 * CERT).
```

Rule 2: If $PLACE_A$ is surrounded with mountains and $PLACE_B$ is near $PLACE_A$ and temperature is normal in $PLACE_A$ with certainty of $CERT$, than the temperature is cold in $PLACE_B$ but with lower certainty say like $0.8 * CERT$.

```
ctx(temperature, cold, NEW_CERT) * ctx(place, PLACE_B,
   1) :-
rel(PLACE_A, surroundedWith, mountains, 1),
rel(PLACE_B, near, PLACE_A, _),
(ctx(temperature, normal, CERT) * ctx(place, PLACE_A,
   1)),
NEW_CERT is (0.8*CERT).
```

Facts:

```
rel('Ile-de-France', locatedIn, 'France', 1).
rel('Paris', locatedIn, 'Ile-de-France', 1).
rel('Sorbonne-Univeristy', locatedIn,'Paris',1).
rel('Versailles', near, 'Paris', 0.7).
rel('Paris', surroundedWith, mountains, 1).
ctx(temperature, normal, 0.9) * ctx(place, 'France', 1).
```

Results:

```
ctx(temperature: normal, 0.90), ctx(place:'France')
ctx(temperature: normal, 0.81), ctx(place:'Ile-de-France')
ctx(temperature: normal, 0.72), ctx(place:'Paris')
ctx(temperature: normal, 0.65),
   ctx(place:'Sorbonne-Univeristy')
ctx(temperature: cold,   0.65), ctx(place:'Versailles'}
```

Remarkably, by defining relations with axioms, the temperature in one place allows us to conclude the temperature in the places related to.

We can increase, decrease and make decisions depending on the value of certainty for context information. Moreover, using different relation properties like reflexive and transitive leads to many results about the same context information. In case many concluded assumptions give the same value with different degrees of certainty, an average function can be applied to produce a more accurate result. Otherwise, when some of these results are inconsistent assuming that the defined rules are reliable and do not produce contradictions, it is possible to identify the wrong assumptions.

5 USE CASE

5.1 Physical Implementation: E-HoA on a Robotic Platform

The E-HoA agent can be implemented on many robotic platforms. For the purpose of validation, we used the widely available Robotis Turtlebot3 Burger[1]. TurtleBot3 is a small, affordable, programmable, ROS-based mobile robot for use in education, research, hobby, and product prototyping. It is composed like a layered infrastructure with spacers that can host the different electronic and mechanical parts, such as the continuous servo-motors with encoders for accurate ground movements, a board for controlling these motors and acquiring measures from different sensors (OpenCR for Turtlebot3), and a Raspberry PI 3 Model B[2] on which runs the ROS framework[3]. Connected to this board, a LIDAR continuously scans the surrounding walls and obstacles. In addition to this standard Turtlebot3 setup, we have added two complementary cameras, one connected to the Raspberry PI 3 for road tracking, and an independent IP camera for observation and detection of objects of interest. The stream captured by the wireless cameras can be transmitted to an off-vehicle GPU-equipped device, an Nvidia Jetson AGX Xavier, for processing object recognition and tracing.

[1] https://emanual.robotis.com/docs/en/platform/turtlebot3/overview/

[2] https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

[3] https://www.ros.org/

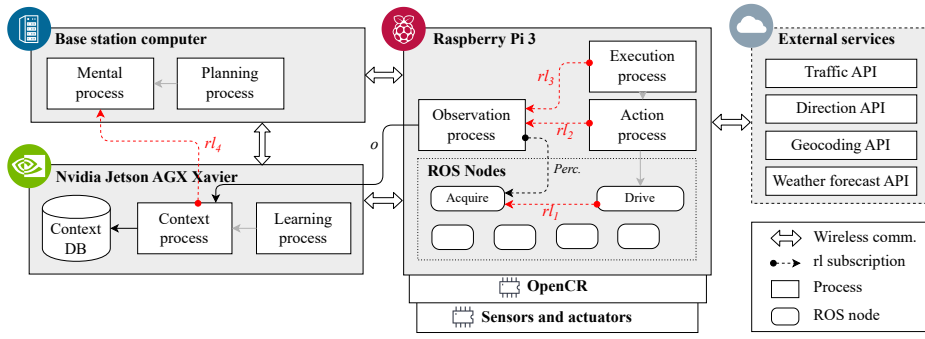


Figure 8. A possible hardware deployment for E-HoA architecture

5.2 Software Considerations: Mapping the E-HoA Processes to Computing Resources

From a system process viewpoint, the E-HoA agent is nothing more than a reduced set of communicating processes that need to be mapped on available computing resources for efficient execution. Due to the demanding amount of computer resources needed by certain tasks (for instance, the observation process requires efficient image processing), some processes or subprocesses may be distributed to computing resources off-vehicle, such as running Convolutional Neural Network software. Also, external services provided in the cloud can be useful to adapt the robot behavior. Figure 8 highlights how the four layers of the E-HoA architecture can be efficiently distributed other a hardware:

- The Mental and Planning processes support the BDI information for mobile applications. Put on the same computer, the Mental process can easily request one or possibly several occurrences of the Planning process to finally develop an execution plan solution.
- The Learning process supported by the Context process relies on an efficient management of history, experience and road map data. For that purpose, we took advantage of the Neo4j graph-oriented database[4], that is particularly good at handling consistency of the acquired spatio-temporal data. This NoSQL Database Management System (DBMS) has been selected because of its intrinsic ability to represent relevant history, experience, map and execution plan with a simple paradigm, nodes containing properties and connected by relations also having properties. This way, experiences are not only spatially specified but also temporally, thus defining a global spatio-temporal context for each experience. The Neo4j DBMS can even run on the Nvidia Jetson AGX Xavier card with noticeable performance that brings great flexibility in the actual mapping of E-HoA processes onto computing resources.

[4] `https://neo4j.com/`

- The execution chain, from the Execution and Action processes to the various ROS nodes, is deployed on a Raspberry Pi 3 directly embedded on the Turtlebot3, so that all the operational activities of the guidance mechanism are concentrated on a single card. The Observation process also runs on the same card, which reveals an efficient way to feedback the environmental information in symbolic terms for playing the nominal and reactive routines.

Choosing ROS as the underlying operating system for E-HoA is natural for development simplicity and portability. Thanks to ROS-compatible tools like Gazebo[5], the E-HoA vehicle and its operating environment can globally be modeled and simulated in 3D, prior to any physical implementation. Once the Gazebo graphical simulation running the E-HoA agent operates correctly, a standard ROS methodology exists that allows to seamlessly shift from virtual simulation to a real physical vehicle operating in a real full-fledged environment. A meta-tool named E-HoA editor has specifically been designed to encapsulate Gazebo in order to automatically perform the correct-by-construction and parameterized procedural generation of scalable use cases.

5.3 Motivational Example: Pharmacy Drug Delivery with Opportunistic Situations

We consider an imaginary city with pharmacies situated in an urban region. Pharmacies handle client prescriptions and transmit the corresponding drug orders to the drug deposit when they do not have the prescribed drugs in their local stock. The drug deposit receives a list of orders/intentions coming from different pharmacies and enjoins an autonomous vehicle to deliver the prescriptions to the appropriate pharmacies, on a daily basis. The targeted vehicle is an electrical autonomous robot equipped with all the previously described features. Assuming a daily order per each pharmacy, the Mental process of the E-HoA agent for contextual execution is helped by the Planning and Learning processes to select the best road sections to schedule the deliveries in a daily tour. The Execution process is then in charge of supervising this daily tour while the Action process can control the execution of each underlying symbolic action, mainly some move operations targeting pharmacies, to be converted on maneuvers over the road map. These processes may delegate to the context process the checking of contextual constraints put on the execution of actions, from those directly solved through the neo4j request language to the more complex prolog-based logical formulas.

It may appear that the delivery truck somewhere can suffer from an event *"battery low"*, due to unexpected traffic jam in some sections or cross-sections of the city map. As the Action process is the single one specified to react at level rl_2, it has subscribed to the Observation process to be informed about this kind of event. Once detected, the current move is stopped by the Action process, so that to be

[5] `http://gazebosim.org/`

replaced by a move to the closest garage, as precised by the Learning process. Once refueled, the Action and Execution processes can offer different ways to resume the daily tour. Thus, the Action process helped by the Learning process could positively evaluate a new series of maneuvers to reach in time the target of the move currently stopped. On the contrary as a service result, the Action process must inform of the failure the Execution process which must try in its turn to find a way to finish the remaining actions in the tour, from the garage location.

It is worth noting that the more 'low' in the layers is the event taken into account the more efficient is the reaction. In particular, when the computation of a totally new execution plan is finally required due to the fact the execution fails to maintain the remaining current one, this should require a deliberation by the Mental process and a heavy activity of the Planning process over a set of intentions.

6 DISCUSSION

The design of context aware systems is an active area of research and a major challenge from raw data acquisition, context modeling and reasoning. Many frameworks, toolkits and middlewares tried to overcome various challenges, like [23, 24].

For the acquisition of context information, diverse models were proposed in [11] from the direct access to sensors to complex proxy middleware. For the sake of robustness and availability, the E-HoA architecture privileges an hybridization of the context information models; The Acquire ROS node implements direct access to field values while the Context process implements both a basic synchronous service and a (asynchronous) blackboard data-centric approaches.

When developing a context-aware system, the choice of a context information model is a corner point since this has impacts on the complexity of context-aware applications, their maintainability and evolvability. Existing approaches vary from the very simple models, which support basic reasoning algorithms that could be deployed in limited use cases, to the powerful ones supporting sophisticated reasoning [13]. In [14], six types of models are mentioned: key-value models, markup scheme models, graphical models, object oriented models, logic-based models and ontological models. The authors conclude that the ontological models are the most promising for the reasoning requirements. But according to [13], an ontological model taken alone is generally unsuited for the recognition of even simple context data. Data cleaning operations and statistical machine learning methods are often required.

Other works promote logic-based modeling. The claim in [25] is that logic approaches appear to be the more expressive although requiring much effort of standardization to improve their re-usability and applicability. Context information is introduced as an abstract mathematical entity with useful additional properties adapted to the artificial intelligence topics. Mainly, an additional relation named $ist(c; p)$ is introduced to assert that the proposition p is true regarding some context c, and a recipe is formalized to perform context lifting from that.

In [26], first order logic predicates are used to describe typed context information. The authors enhance the interest of type checking, considering for instance $Location(Chris, entering, room3231)$ as a typed element, where the first argument must be a person or an object. In [27], a first order logic is used to define context information in a more generic and consistent way, introducing one predicate to model whatever $context\_element(entity, state/value, time)$.

The E-HoA software architecture also privileges an hybrid approach to model and manage the context information, as a third-part approach:

- The Acquire ROS node acts as a preprocessor to clean up input context data.
- With respect to the ontology representing the context information, some typed data and relations are specified in a based-graph database (Neo4j) which is known to be scalable, able to handle huge datasets.
- As the Context process handles the former database, it is able to perform even complex requests on context data and on their relations. In fact, there are three ways yielding values of context data:

  1. The Context process can serve any other E-HoA process subscribing to the truth of some context-based logic formula in order to react to the possible changes on-the-fly;
  2. The Context process can request some machine learning process, e.g. to compute mean traffic information according to some spatio-temporal constraints [18];
  3. And as stated in this paper, context data can also be evaluated with a degree of certainty due to the specification of context data relations.

7 CONCLUSION

Dedicated to context-aware autonomous vehicles, the E-HoA scalable multi-process architecture combines deliberative intentional concepts and reactive capabilities. From the observed events, it is used to guide the vehicle to satisfy some sets of intentions, while adapting its behavior under the dynamic environmental circumstances. The proposed functionalities can be adapted to whatever vehicles which run the known ROS system.

With respect to the existing layered architectures, all the E-HoA processes are context-centric and are supervised by a context layer which handles both concrete/symbolic information and offer estimation services based on previously learnt experiments.

In order to improve the relevance of the contextual information that can be deduced from the observed data, we have investigated the way to formalize the acquisition of contextual information in three successive stages:

1. The acquisition of the raw contextual information (sensors) comes with a low level data processor exploiting noise filtering and frequency control;

2. The symbolization mechanism based on fuzzy logic, it helps certifying the context data properties;
3. The context reasoning introduces first order logic to make the higher level information emerge.

Furthermore, thanks to the correct handling of four reactivity levels (from arch-reflex to intent-reflex), intentions (globally converted into an optimized sequence of atomic vehicle maneuvers) and events can be tightly and consistently intertwined as the spatio-temporal context evolves, and the computed execution plan can accordingly be updated in real-time.

Although promising, we consider this work as a foundation. Concrete benchmarking approaches are required to evaluate the proposed observation mechanisms and measure its impact on the dynamic of the vehicle behavior. An immediate perspective of improvement would consist in pushing deeper machine learning techniques on data observation, both to help configuring the deduction system and to discover alternative opportunities of actions.

REFERENCES

[1] Van Brummelen, J.—O'Brien, M.—Gruyer, D.—Najjaran, H.: Autonomous Vehicle Perception: The Technology of Today and Tomorrow. Transportation Research Part C: Emerging Technologies, Vol. 89, 2018, pp. 384–406.

[2] Long, L. N.—Hanford, S. D.—Janrathitikarn, O.—Sinsley, G. L.—Miller, J. A.: A Review of Intelligent Systems Software for Autonomous Vehicles. 2007 IEEE Symposium on Computational Intelligence in Security and Defense Applications, IEEE, 2007, pp. 69–76, doi: 10.1109/CISDA.2007.368137.

[3] Fróes, E.—Gudwin, R. R.: Building a Motivational Subsystem for the Cognitive Systems Toolkit. SCASBA, 2017, pp. 1880–1886.

[4] Gudwin, R.—Paraense, A.—de Paula, S. M.—Fróes, E.—Gibaut, W.—Castro, E.—Figueiredo, V.—Raizer, K.: The Multipurpose Enhanced Cognitive Architecture (MECA). Biologically Inspired Cognitive Architectures, Vol. 22, 2017, pp. 20–34.

[5] Paraense, A. L.—Raizer, K.—de Paula, S. M.—Rohmer, E.—Gudwin, R. R.: The Cognitive Systems Toolkit and the CST Reference Cognitive Architecture. Biologically Inspired Cognitive Architectures, Vol. 17, 2016, pp. 32–48.

[6] Bordini, R. H.—Dastani, M.—Dix, J.—El Fallah Seghrouchni, A.: Multi-Agent Programming. Springer, 2009, doi: 10.1007/978-0-387-89299-3.

[7] Ziafati, P.—Dastani, M.—Meyer, J.—van der Torre, L.: Event-Processing in Autonomous Robot Programming. AAMAS'13, 2013, pp. 95–102.

[8] Kortenkamp, D.—Simmons, R.—Brugali, D.: Robotic Systems Architectures and Programming. Springer Handbook of Robotics, Springer, 2016, pp. 283–306, doi: 10.1007/978-3-319-32552-1_12.

[9] Alzetta, F.—Giorgini, P.: Towards a Real-Time BDI Model for ROS 2. Proceedings of the 20th Workshop From Objects to Agents, Parma, Italy, June 26th–28th, 2019, 2019, pp. 1–7.

[10] Martinez, A.—Fernndez, E.: Learning ROS for Robotics Programming. Packt Publishing, 2013.

[11] Baldauf, M.—Dustdar, S.—Rosenberg, F.: A Survey on Context-Aware Systems. International Journal of Ad Hoc and Ubiquitous Computing, Vol. 2, 2007, No. 4, pp. 263–277, doi: 10.1504/IJAHUC.2007.014070.

[12] Loke, S.: Context-Aware Pervasive Systems: Architectures for a New Breed of Applications. Auerbach Publications, 2007.

[13] Bettini, C.—Brdiczka, O.—Henricksen, K.—Indulska, J.—Nicklas, D.—Ranganathan, A.—Riboni, D.: A Survey of Context Modelling and Reasoning Techniques. Pervasive and Mobile Computing, Vol. 6, 2010, No. 2, pp. 161–180, doi: 10.1016/j.pmcj.2009.06.002.

[14] Strang, T.—Linnhoff-Popien, C.: A Context Modeling Survey. Workshop on Advanced Context Modeling, Reasoning and Management as Part of Ubicomp, 2004.

[15] Sathe, S.—Papaioannou, T. G.—Jeung, H.—Aberer, K.: A Survey of Model-Based Sensor Data Acquisition and Management. In: Aggarwal, C. C. (Ed.): Managing and Mining Sensor Data. Springer US, Boston, MA, 2013, pp. 9–50, doi: 10.1007/978-1-4614-6309-2_2.

[16] Ilié, J. M.—Chaouche, A. C.—Pêcheux, F.: E-HoA: A Distributed Layered Architecture for Context-Aware Autonomous Vehicles. Procedia Computer Science, Vol. 170, 2020, pp. 530–538, doi: 10.1016/j.procs.2020.03.121.

[17] Chaouche, A. C.—El Fallah Seghrouchni, A.—Ilié, J. M.—Saïdouni, D. E.: A Higher-Order Agent Model with Contextual Management for Ambient Systems. TCCI XVI, Springer Berlin Heidelberg, LNCS, Vol. 8780, 2014, pp. 146–169, doi: 10.1007/978-3-662-44871-7_6.

[18] Ilié, J. M.—Chaouche, A. C.—Pêcheux, F.: A Reinforcement Learning Integrating Distributed Caches for Contextual Road Navigation. International Journal of Ambient Computing and Intelligence (IJACI), Vol. 13, 2022, No. 1, pp. 1–19, doi: 10.4018/IJACI.300792.

[19] Chaouche, A. C.—Ilié, J. M.—Pêcheux, F.: Dealing with Failures for Execution Consistency in Context-Aware Systems. Vol. 177, 2020, pp. 212–219, doi: 10.1016/j.procs.2020.10.030.

[20] Schafer, R. W.: What Is a Savitzky-Golay Filter? [Lecture Notes]. IEEE Signal Processing Magazine, Vol. 28, 2011, No. 4, pp. 111–117.

[21] Redmon, J.—Farhadi, A.: YOLOv3: An Incremental Improvement. 2018, arXiv: 1804.02767.

[22] Wielemaker, J.—Schrijvers, T.—Triska, M.—Lager, T.: SWI-Prolog. 2010, arXiv: 1011.5332.

[23] Park, J.—Moon, M.—Hwang, S.—Yeom, K.: Cass: A Context-Aware Simulation System for Smart Home. 5th ACIS International Conference on Software Engineering Research, Management Applications (SERA 2007), 2007, pp. 461–467.

[24] Zeynalvand, L.—Luo, T.—Zhang, J.: COBRA: Context-Aware Bernoulli Neural Networks for Reputation Assessment. 2019, arXiv: 1912.08446.

[25] McCarthy, J.—Buvac, S.: Formalizing Context (Expanded Notes). Technical Report. Stanford University, Stanford, CA, USA, 1994.

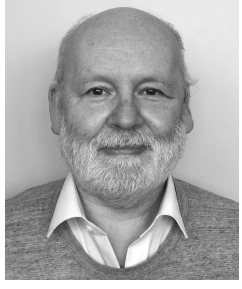
[26] Ranganathan, A.—Campbell, R. H.: An Infrastructure for Context-Awareness Based on First Order Logic. Personal and Ubiquitous Computing, Vol. 7, 2003, No. 6, pp. 353–364.

[27] Miraoui, M.—El-Etriby, S.—Abed, A. Z.—Tadj, C.: A Logic Based Context Modeling and Context-Aware Services Adaptation for a Smart Office. International Journal of Advanced Studies in Computers, Science and Engineering; Gothenburg, Vol. 5, 2016, pp. 1–6.

Ahmed-Chawki Chaouche has received his Ph.D. in computer science from both Sorbonne University (former UPMC) in France and University of Abdelhamid Mehri, Constantine 2 in Algeria (2015). Currently, he is an Associate Professor at the Constantine 2 University. He is also a Permanent Researcher in computer science and Accredited Research Director at the MISC Laboratory. His research interests include ambient intelligence systems, autonomous vehicles, implementation of IoT and connected objects, planning mechanisms and learning approaches.

Jean-Michel Ilié obtained several degrees in electronics and informatics among with the Ph.D. thesis from the University Pierre and Marie Curie in France (1990). Currently, a member of the Paris City University in its conference master higher grade (2009), he is also a Permanent Researcher of the LIP6 laboratory at the Sorbonne University. The fields of his research concern the formal validation of complex embedded distributed systems and the emergence of adapted behaviours when coping with dynamic contexts. In the last 15 years, he has tackled the way to define intelligent software agents in complex ambient systems for autonomous activity. His research keywords include spatio-temporal planning, autonomous guidance, intelligent transportation.

Assem Hebik received his Master's degree in science and technologies of information and communication (2020) from the University of Constantine 2 with excellence. Previously he had obtained a License degree with the first class honors from the same university. Currently, he has been Top rated plus freelancer on Upwork for more than three years during which he had the chance to work with international software development teams across the globe. His primary focus is scientific research, that is why he volunteers with research teams when he gets a chance.



François Pêcheux is Full Professor at the Sorbonne Université, Paris, France. He is currently heading the Polytech Sorbonne Engineering School. His research activities focus on the modelling and simulation of digital-centric heterogeneous systems. He participated in the development of numerous CAD tools for electronic design automation, especially event-driven and analogue simulators. He published more than 80 journal and conference papers in this domain.

Computing and Informatics, Vol. 42, 2023, 741–761, doi: 10.31577/cai_2023_3_741

EEG-EMG ANALYSIS METHOD IN HYBRID BRAIN COMPUTER INTERFACE FOR HAND REHABILITATION TRAINING

Lubo Fu, Haoyang Li, Hongfei Ji*

Department of Computer Science and Technology
School of Electronic and Information Engineering
Tongji University, Shanghai 201804, China
e-mail: {2130758, haoyangli, jhf}@tongji.edu.cn

Jie Li*

Translational Research Center
Shanghai YangZhi Rehabilitation Hospital
(Shanghai Sunshine Rehabilitation Center)
Tongji University
&
Department of Computer Science and Technology
School of Electronic and Information Engineering
Tongji University, Shanghai 201804, China
e-mail: nijanice@163.com

Abstract. Brain-computer interfaces (BCIs) have demonstrated immense potential in aiding stroke patients during their physical rehabilitation journey. By reshaping the neural circuits connecting the patient's brain and limbs, these interfaces contribute to the restoration of motor functions, ultimately leading to a significant improvement in the patient's overall quality of life. However, the current BCI primarily relies on Electroencephalogram (EEG) motor imagery (MI), which has relatively coarse recognition granularity and struggles to accurately recognize specific hand movements. To address this limitation, this paper proposes a hybrid BCI framework based on Electroencephalogram and Electromyography (EEG-

* Corresponding author

EMG). The framework utilizes a combination of techniques: decoding EEG by using Graph Convolutional LSTM Networks (GCN-LSTM) to recognize the subject's motion intention, and decoding EMG by using a convolutional neural network (CNN) to accurately identify hand movements. In EEG decoding, the correlation between channels is calculated using Standardized Permutation Mutual Information (SPMI), and the decoding process is further explained by analyzing the correlation matrix. In EMG decoding, experiments are conducted on two task paradigms, both achieving promising results. The proposed framework is validated using the publicly available WAL-EEG-GAL (Wearable interfaces for hand function recovery Electroencephalography Grasp-And-Lift) dataset, where the average classification accuracies of EEG and EMG are 0.892 and 0.954, respectively. This research aims to establish an efficient and user-friendly EEG-EMG hybrid BCI, thereby facilitating the hand rehabilitation training of stroke patients.

Keywords: Hybrid BCI, EEG, EMG, GCN, neural networks

1 INTRODUCTION

Stroke is a debilitating condition caused by the blockage or rupture of blood vessels, resulting in damage to brain cells. It often leads to various neurological deficits, including unilateral paralysis, cognitive impairment, and language difficulties. Among the challenges faced by stroke survivors, upper limb impairment significantly impacts their ability to perform essential activities of daily living (ADLs) such as eating, dressing, and personal hygiene. Given the intricate and precise movements required for these tasks, effective hand rehabilitation is crucial to restore patients' independence in performing these fundamental activities [1, 2].

1.1 Rehabilitation Training Based on EEG MI

Motor imagery (MI) refers to the mental process of envisioning movement without actually physically executing it [3, 4, 5, 6, 7]. It has been widely utilized by both healthy individuals for learning new movement skills during exercise [8] and stroke patients during rehabilitation training [9]. The underlying principle behind MI lies in the activation of brain regions within the sensorimotor network [10]. Thus, for patients facing difficulties in performing physical movements during rehabilitation, MI can be employed to activate partially damaged motor networks, aiding them in the gradual restoration of movement [11]. Numerous studies have demonstrated the effectiveness of EEG-based MI in rehabilitation.

EEG recordings are obtained by measuring the potential between a signal electrode and a reference electrode placed on the scalp, which is easily contaminated by eye and muscle movement. Furthermore, EEG exhibits limitations in spatial resolution, typically ranging from 5 to 9 centimeters [12], and it can only capture

neuronal population potentials in broad brain regions. As a result, EEG is primarily capable of detecting coarse-grained changes in brain signals, often unable to discern the finer and more intricate movements associated with the affected limb. Thus, relying solely on EEG poses challenges in perceiving and capturing the complexities of movement.

1.2 Dynamic Graph Convolutional Networks for BCIs

Traditionally, EEG decoding has involved processing data from each channel independently, without considering the inter-channel correlations. However, by treating EEG as graph-structured data, it becomes possible to leverage the relationships between channels and achieve more comprehensive EEG decoding. One approach to handling graph-structured data is to use the graph convolutional networks (GCN) [13]. Notably, Song et al. successfully applied GCN to EEG emotion recognition in 2018, yielding promising outcomes [14].

To address the challenge of limited EEG data volume, Zhang et al. proposed GCB-net [15]. GCB-net utilizes graph convolution layers to explore the correlations between EEG channels and employs the broad learning system (BLS) mechanism to map the extracted features into a wider feature space, resulting in enhanced robustness. Moreover, to further uncover the relationships between EEG channels, dynamic graph convolution has gained significant traction [16]. Dynamic graph neural networks employ a learnable adjacency matrix as a parameter, which is updated during the training process [17, 18, 19].

In this study, a similar approach is adopted, where graph convolution is employed to capture the correlations between EEG channels. Additionally, LSTM is utilized to address the temporal dynamics inherent in EEG signals.

1.3 EEG-EMG-Based Hybrid BCIs

EMG, obtained by recording the electrical activity of skeletal muscles through surface sensors, possesses notable advantages over EEG. It exhibits good stability, high signal strength, and the ability to discern finer body movements in healthy individuals. Many studies [20, 21, 22, 13, 23] have demonstrated that EMG-based techniques can achieve high accuracy in multi-gesture recognition with fewer leads and shorter calibration times.

Research on the EEG-EMG-based Hybrid BCIs has already been initiated. Leeb et al. [24] conducted a fusion study using EEG and EMG signals to enhance the classification accuracy of MI. Lin et al. [25] combined visually evoked potentials (SSVEP) with EMG to increase the number of targets and improve information transmission rates. Sarasola-Sanz et al. [26] employed EEG and EMG to control a mechanical exoskeleton, enabling control of a seven-degree-of-freedom robotic arm. Some studies have explored the coupling of EEG and EMG signals. Tun et al. [27, 28] investigated the functional coupling between EEG and EMG during four distinct movements. Soundirarajan et al. [29] evaluated the coupled responses

of facial muscles and the brain to various motor visual stimuli by analyzing the information embedded in EEG and EMG signals.

In this study, the participants' active intentions are captured through EEG, utilizing EEG decoding to monitor their motor intentions. Additionally, leveraging the fine-grained classification capability of EMG, action recognition is achieved through EMG decoding.

2 MODEL FRAMEWORK

This section introduces the comprehensive framework employed in this study, depicted in Figure 1. The framework utilizes both EEG and EMG signals for hand rehabilitation training. Firstly, the EEG signals are decoded to detect the user's intended movements. Subsequently, the decoded intention is used to guide the decoding of the EMG signals, facilitating the classification of specific hand actions. External devices are employed to provide additional support for the rehabilitation training process.

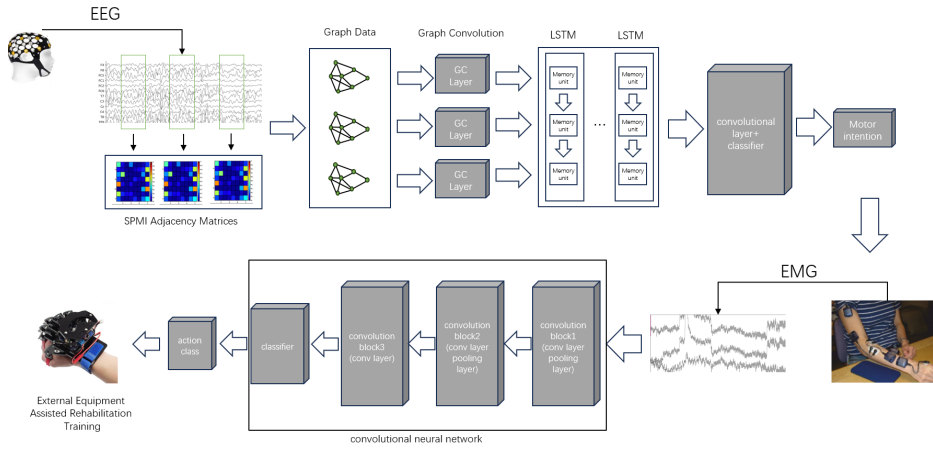


Figure 1. Overall framework: After the EEG signal is obtained through the device, it is processed into serialized graph structure data, and then processed by GCN-LSTM to detect motion intention. Then, for the obtained EMG, we use CNN to decode and realize action recognition, so as to help the subjects to carry out hand rehabilitation training.

We commence by acquiring 32-channel EEG data through the utilization of an EEG cap. Subsequently, the EEG data are segmented into four segments. For each segment, the pairwise SPMI between each channel is calculated, resulting in the construction of a relational adjacency matrix. This matrix facilitates the creation of a serialized graph structure representation of the EEG data.

Next, the decoding process begins by employing a graph convolutional (GC) layer for each segment. Additionally, for each vertex, all its corresponding segments

form a sequence, which is then processed using a Long Short-Term Memory (LSTM) network. To enable deeper decoding, a convolutional block is applied, followed by a classification layer that generates predictions regarding the user's motion intentions.

Simultaneously, the EMG data is processed using a CNN composed of three convolutional blocks. The EMG decoding primarily focuses on extracting relevant information from the temporal dimension.

3 MATERIALS AND METHODS

3.1 Data Description

We utilize the publicly available WAL-EEG-GAL dataset [30] for our study. This dataset captures simultaneous EEG and EMG recordings from 12 subjects while they perform repetitive grasping and lifting trials. Each subject participates in several series, and each series consists of 34 repeated trials.

During each trial, the participants are instructed to reach out and grasp a small object using their thumb and forefinger, lift it into the air, hold it for a few seconds, and then lower it back to its initial position. The entire process lasts approximately 8 seconds, with LED indicators used to signal the lifting and lowering phases, while other aspects of the rhythm are controlled by the participants themselves. A total of 32 electrodes are used to record the EEG signals, while 5 electrodes are employed for EMG recordings. The EEG signals are sampled at a frequency of 500 Hz, and the EMG signals are sampled at 4 000 Hz.

Across different series, the weight of the grasped object (150 g, 300 g, 600 g) and the surface material (sandpaper, suede, silk) varied. However, for our study, we focus solely on the series with object weight variations, while ensuring that the surface material remained consistent (sandpaper).

As shown in Figure 2, the upper two figures respectively represent the schematic diagram of the EEG channel and the schematic diagram of the EMG channel. Among them, for EEG, we adopt the international standard 10-20 system, and use 1–32 channels as shown in the figure. For EMG, we use 5 positions on the arm: the anterior deltoid (AD), brachioradial (BR), flexor digitorum (FD), common extensor digitorum (CED), and the first dorsal interosseus muscles (FDI). The scale below indicates the key time points in a trial process, and a trial lasts 8 s. Based on the actions performed at each time point and whether the object is touched, the trial can be divided into two distinct stages. The first stage, lasting from 0 to 4 seconds, represents the initial stage of movement. The subsequent stage, spanning from 4 to 8 seconds, corresponds to the specific execution stage of the movement. More specifically, the period from 0 to 2 seconds represents the resting stage, while the interval from 2 to 4 seconds corresponds to the action stage. The EEG signals recorded during these two stages can be analyzed to enable the model to recognize the intended movement.

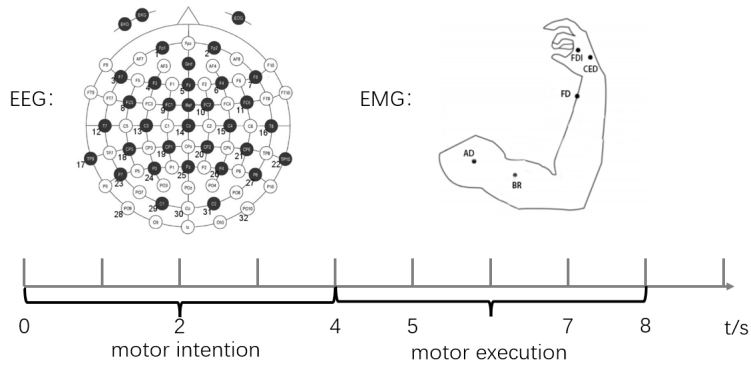


Figure 2. An introduction to each time point of a trial, 2 s: the LED light is on, indicating that the subject starts to move; 2–4 s: the subject reaches for the object; 4–5 s: the object leaves the table; 8 s: the object is put back on the table. We use 0–4 s EEG data to detect motion intention, and 4–8 s EMG data to classify motion execution.

3.2 Data Processing

3.2.1 EEG Processing

The original EEG data is represented as $X \in \mathbb{R}^{c \times t}$, where $\mathbf{c}$ denotes the number of channels ($\mathbf{c} = 32$), and $\mathbf{t}$ represents the number of sampling points ($\mathbf{t} = 4\,000 = 8 \times 500$). In this study, the EEG data of the first 4 s is used for the detection of motion intention. Specifically, the data from 0 to 2 seconds is assigned as class 0, while the data from 2 to 4 seconds is assigned as class 1. Consequently, the value of $\mathbf{t}$ is reduced to 1 000 (2×500). To streamline the computational load, we downsample the EEG data by reducing its frequency to half of the original. Additionally, to eliminate noise and extract signals relevant to motion classification, a band-pass filter with a range of 4 to 35 Hz is applied to the EEG data.

EEG data represents a time-series signal. To fully leverage its temporal characteristics, the EEG signal of a trial is partitioned into T segments (in this study, T is set to 4), resulting in a data representation of $(X_i)_{i \in \mathbb{Z}_T}$, where $\mathbb{Z}_T := \{1, 2, \ldots, T\}$. Additionally, EEG comprises multiple channels of data, and there exists a certain interrelation between these channels. By treating EEG as graph-structured data, we can explore the relationships between channels and comprehensively analyze the EEG signal.

For each step i, each channel of EEG is treated as a vertex v_i. By calculating SPMI, we derive the connections e_i between channels, leading to the generation of an adjacency matrix A_i. Consequently, undirected graphs $G_i = (V_i, E_i)$ are formed. The data corresponding to each channel serves as the feature vector for the respective vertex X_i.

3.2.2 EMG Processing

The original EMG data is represented as $Y \in \mathbb{R}^{c \times t}$, where $\mathbf{c}$ denotes the number of channels ($\mathbf{c} = 5$), and $\mathbf{t}$ represents the number of sampling points ($t = 32\,000 = 8 \times 4\,000$). In this study, the EMG data from the last 4 seconds is utilized for classification, resulting in $\mathbf{t}$ being equal to 16 000 ($4 \times 4\,000$).

To begin with, the EMG signal is downsampled from 4 000 Hz to 250 Hz in order to reduce the sampling frequency. Subsequently, a filtering process is applied to the signal within the frequency range of 0 Hz to 100 Hz to remove unwanted frequencies and retain the relevant information for further analysis.

3.3 Classification Methods

In this section, two primary models are introduced for processing EEG and EMG signals, respectively. The GCN-LSTM is utilized to handle the EEG data, while the CNN is employed to process the EMG data.

3.3.1 EEG Classification Based on GCN-LSTM

We use a GCN-LSTM to process EEG, which consists of the following two components:

- GCN: Graph convolution is capable of handling graph-structured data, allowing for the processing of feature information for each vertex while considering the connections between vertices. However, it does not possess the ability to handle time series information.
- LSTM: LSTM facilitates the backward propagation of time series information through its memory unit, making it advantageous for handling time series data. However, it may not effectively utilize the connection relationships between vertices in graph-structured data.

We leverage the strengths of both GCN and LSTM to construct a GCN-LSTM for EEG processing, as shown in Figure 3.

For EEG, we preprocess it into serialized graph structured data. Here, each channel of the EEG is represented as a vertex on the graph, and the relationship between the vectors and among the channels constitutes the adjacency matrix of the graph. Let $(G_i)_{i \in \mathbb{Z}_T}$ with $\mathbb{Z}_T := \{1, 2, \ldots, T\}$ represent a finite sequence of undirected graphs $G_i = (V_i, E_i)$, where $V_i \in V \ \forall i \in \mathbb{Z}_T$. All graphs in the sequence share the same set of vertices, but the vertex feature vectors and adjacency matrices may differ among the graphs.

In this study, SPMI [31] is used to calculate the connection between vertices, so as to obtain the adjacency matrix A_i. For two channel vectors X and Y of a signal, their correlation can be calculated as follows.

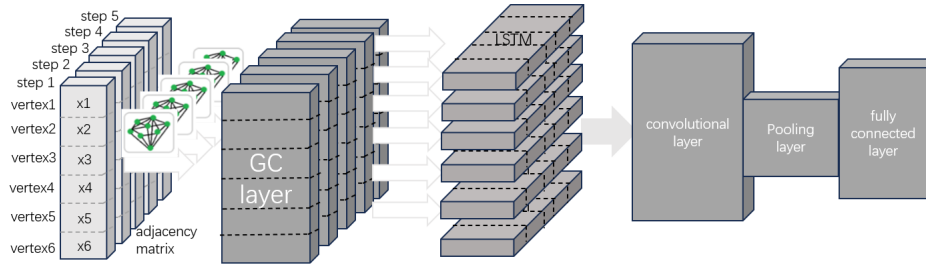


Figure 3. GCN-LSTM: A model for classifying serialized graph-structured EEG, consisting of GC layers, LSTM layers, and convolutional blocks

First, calculate the permutation entropy of the vector X, as follows:

$$PE_X = -\sum_{i=1}^{n!} P_X(i)\log(P_X(i)), \tag{1}$$

where $P_X(i)$ is the empirical probability of the i^{th} ordered pattern of X, and n is the dimension of X. Then the joint PE of signals X and Y is defined as follows:

$$PE_{X,Y} = -\sum_{i=1}^{n!}\sum_{j=1}^{n!} P_{X,Y}(i,j)\log(P_{X,Y}(i,j)), \tag{2}$$

where $P_{X,Y}(i,j)$ is the joint probability of permutation of X and Y. Finally the *SMPI* of X and Y can be calculated as follows:

$$SPMI_{X,Y} = \frac{PE_X + PE_Y - PE_{X,Y}}{PE_{X,Y}}. \tag{3}$$

As depicted in the figure, the serialized graph-structured EEG data can be segmented into **T** steps. To decode each step, we utilize a GC layer, and a total of T parallel GC layers are employed to process all T steps. Specifically, at step **i**, the vertex feature vector set $X_i^0 \in \mathbb{R}^{|V|\times d}$ serves as the input to the GCN layer. The adjacency matrix A_i of the graph is employed to aggregate the neighborhood information. Subsequently, a weight matrix $W_i \in \mathbb{R}^{d\times d}$ is applied to update the vertex embedding vector set. The mathematical form of this process can be expressed as follows:

$$\begin{aligned} X_i^1 &= GCL_i(A_i, X_i^0, W_i), \\ &:= \sigma(A_i X_i^0 W_i), \end{aligned} \tag{4}$$

where $X_i^1 \in \mathbb{R}^{c\times d}$, σ is an activation function.

For vertex j, its T steps form a sequence, expressed as $(x_{i,j})_{i\in\{1,2,\ldots,T\}}$. These sequences are then processed by an LSTM layer, with a total of **c** such layers used to

process all **c** vertices. For a given vertex **j**, the output of the LSTM layer is obtained through the following calculation steps.

The first step involves determining which information should be retained or forgotten from the cell state. This decision is governed by the "forget gate" layer, which uses a sigmoid function to determine whether to completely forget or partially retain information from the previous time step. At step i, the calculation can be expressed as follows:

$$f_{i,j} = \sigma(W_f \cdot [h_{i-1,j}, x_{i,j}] + b_f). \tag{5}$$

The second step involves generating new information that we need to incorporate for updating. This step comprises two parts. The first part is an "input gate" layer that utilizes the sigmoid function to determine the values that should be updated. The second part involves a tanh layer that generates new candidate values and combines them together to yield the candidate values. The process can be described as follows:

$$C_{i,j} = f_{i,j} * C_{i-1,j} + m_i * \tilde{C_{i-1,j}}. \tag{6}$$

The final step is to determine the output of the model. Initially, an initial output is obtained through the sigmoid layer. This output is then scaled to a range of -1 to 1 using the tanh function. The scaled output is multiplied element-wise with the output obtained from the sigmoid layer to obtain the final output of the model.

$$o_{i,j} = \sigma(W_o[h_{i-1,j}, x_{i,j}] + b_o), \tag{7}$$

$$h_{i,j} = o_{i,j} * tanh(C_{i,j}). \tag{8}$$

We obtain the hidden state of the last step as the output of the LSTM, so we have

$$x_j^2 = LSTM\left(x_j^1\right). \tag{9}$$

Here, the symbol σ represents the sigmoid function, as illustrated in Equation (9), and tanh denotes the hyperbolic tangent function, as depicted in Equation (10):

$$\sigma(x) = \frac{1}{1+e^{-x}}, \tag{10}$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{11}$$

The corresponding sequence $(x_{i,j})_{i\in\{1,2,\ldots,T\}}$ is transformed into a sequence $x_j^1 \in \mathbb{R}^{d^1}$ representing the embedded feature vector for vertex **j**. In turn, all **c** embedded feature vectors are concatenated into a vertex vector set $X^2 \in \mathbb{R}^{c\times d^2}$. A convolution block is then applied to further decode these features, followed by a fully connected layer and a softmax function for obtaining the final classification probabilities.

$$y = softmax(linear(\sigma(Covn(X^2)). \tag{12}$$

By analyzing the classification results, we can get the results of motion intention detection.

3.3.2 CNN-Based EMG Classification

We propose a novel approach for efficiently decoding EMG signals using a CNN, as illustrated in Figure 4. After collecting EMG on the arm using 5 EMG electrodes, some preprocessing operations are performed on the raw data. We reduce the data dimension by downsampling, remove noise and impurities by filtering, and obtain useful signals. The processed data is then decoded using a CNN. Considering the limited number of EMG channels ($c = 5$), we mainly apply convolutions over the time dimension. The architecture comprises three convolutional blocks, where the first two are composed of a single convolutional layer followed by a max-pooling layer, while the third block utilizes only convolutional layers. The ReLU activation function is utilized throughout the network, and dropout is employed after each convolutional layer to alleviate overfitting. For each convolutional block i, the input is Y^i, which is then processed as follows.

$$Y^{i+1} = MaxPooling(Conv2D(Y^i)). \tag{13}$$

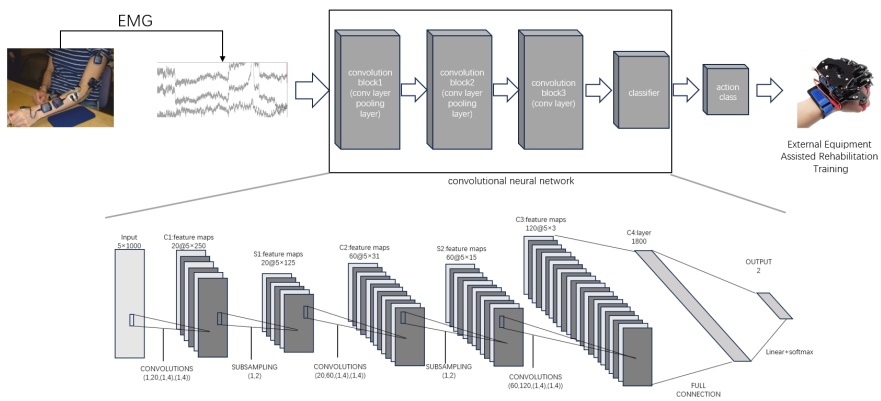


Figure 4. CNN: A convolutional network for EMG classification, consisting of three convolutional blocks, which mainly decodes from the temporal dimension of EMG

Following convolutional blocks, we employ a fully connected layer to further process the extracted features, and subsequently apply a linear layer and softmax function to obtain the classification probability. Specifically, the linear layer computes the weighted sum of the features, and then the softmax function maps the resulting vector to a probability distribution over the classes. This allows for accurate classification of EMG signals with a high degree of confidence.

$$y = Softmax(Linear(Flatten(Y))). \tag{14}$$

After obtaining the classification results of EMG, control signals are sent to the peripheral devices (such as mechanical gloves) connected to the computer to assist

the subject's movement, thereby helping the subject to perform hand rehabilitation training.

4 RESULTS AND DISCUSSION

4.1 Intention Detection

4.1.1 Experiments Settings

We employ the GCN-LSTM model to process the serialized graph structure of EEG signals and detect movement intention. The training of the model consists of 200 epochs with a batch size of 10. To optimize the model, we use the Adam optimizer with a learning rate of 1e−3 and a weight decay parameter of 1e−3. The loss function is implemented as the sum of cross-entropy between the predicted label and the true label. L2 regularization is also applied during the training phase to reduce overfitting. During the evaluation phase, the average accuracy of test data serves as the key metric to assess performance of the model. Our evaluation results demonstrate the effectiveness of the proposed GCN-LSTM approach in accurately decoding EEG signals for estimating movement intention.

The model is evaluated in two ways. On the one hand, the correlation between channels is calculated, and the interpretability is illustrated by analyzing the connection of channels. On the other hand, the validity of the model is verified by the average accuracy rate.

4.1.2 Experiments Results

Our proposed model is evaluated for its accuracy in detecting EEG motion intentions. We conduct experiments with 12 subjects and compared the results with the CSP + SVM model [32] as the baseline. As shown in Figure 5, the experimental results demonstrate that our model outperforms the baseline model across all subjects. Specifically, the accuracy rate for subjects 1, 2, 4, 7, 9, 10, and 11 exceeded 90 %, while the performance for test 5 is suboptimal, achieving only slightly above 70 %. This lower accuracy for test 5 may be attributable to poorer signal quality in that particular experiment.

To further validate the interpretability of our model, we will conduct experiments to analyze its performance. In each trial (0–8 s), we will divide the data into four segments, each consisting of 2 seconds. For each segment, we will calculate the inter-channel correlation using SPMI. Interpretability of our model will be validated through experiments.

Figure 6 shows the channel correlation matrix of the four stages and the corresponding connection visualization. The matrix is represented by 6 a), 6 b), 6 e), and 6 f) corresponding to each stage. The figure is structured horizontally from left to right and vertically from top to bottom. The channel order follows the same sequence as that of channels (1–32) shown on the 10-20 system in Figure 2.

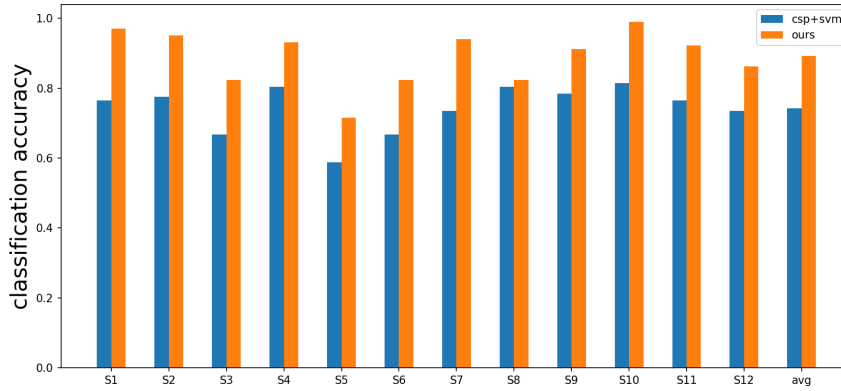
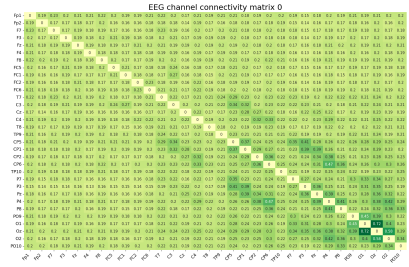


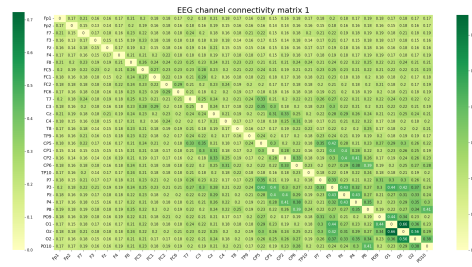
Figure 5. EEG classification accuracy

The number bar on the right side of each figure shows the strength of the relationship between channels. The relationship becomes stronger from bottom to top. Figures 6 c), 6 d), 6 g), and 6 h) are the corresponding connection visualization diagrams. During the creation process, the 20 connections with the highest connection strength are selected, and the connections between the channels with a distance of less than 5 cm are removed to reduce the interference caused by the close distance.

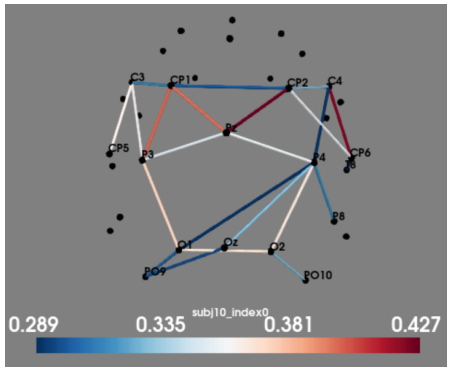
Figure 6 reveals two notable patterns. First, although the overall data corresponds to 32 channels, only about 15 channels exhibit significant connectivity during the motion process. Second, the areas with strong connectivity are motor and visual areas, which are consistent with the form of the task action. Although the connection between the leads changes in the four stages, the channels with strong connectivity remain the same. This indicates that the subject's motor and visual areas remained active throughout this period. To enhance the universality of the channel selection process, we employ a method to refine it further. Initially, we select a representative sample consisting of 12 subjects to undergo the channel selection process. Each subject goes through four distinct stages, thereby generating a total of 48 data pieces. Subsequently, we compute the channel correlation matrix using these data samples. From each correlation matrix, we identify the 15 connections with the highest correlation values. We take out the channels associated with these connections and proceed by tallying the frequency of occurrence for each channel across the 48 sets of data. This frequency count enables us to determine the popularity of each channel within the dataset. Ultimately, to facilitate our experimentation, we select the top 15 channels ('P3', 'P4', 'Pz', 'Oz', 'O2', 'CP1', 'O1', 'CP2', 'CP5', 'CP6', 'P8', 'PO9', 'P7', 'PO10', 'C3') with the highest frequency of occurrence. These channels will be utilized for further analysis and investigation. This suggests that these channels are the most informative.



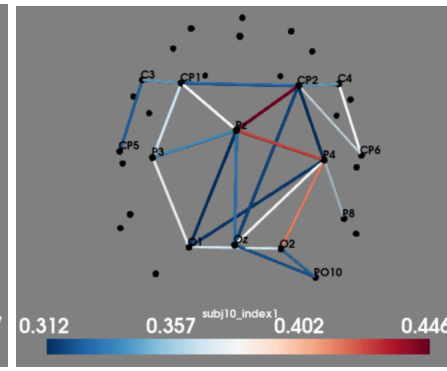
a) 0–2 s



b) 2–4 s



c) 0–2 s



d) 2–4 s

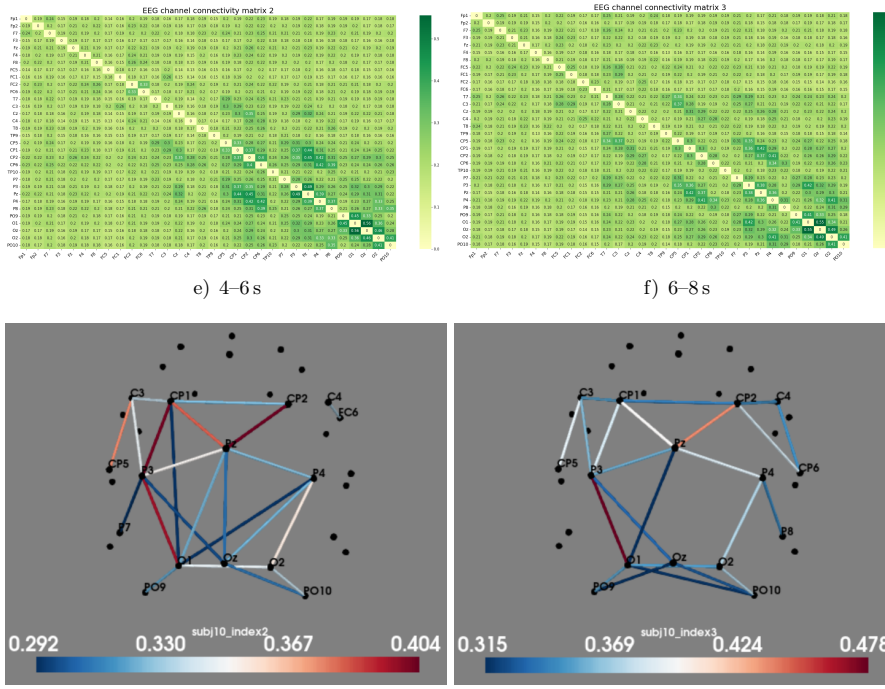
To demonstrate that the well-connected channels provide more effective information, we conduct a series of controlled experiments. Specifically, we limit the data used for motion intent recognition to the 15 channels exhibiting strong connectivity. The results are shown in the Figure 7.

As displayed in Figure 7, despite using data from less than half of the original channels, the accuracy rate did not decrease significantly. Notably, Tests 4 and 11 even achieved results that are equal to or greater than the original 32-channel setup. In terms of average accuracy, the 15-channel configuration is only 0.07 lower than the 32-channel configuration. These findings demonstrate that effective channels can be identified through analyzing channel connectivity. This not only provides an explanation for EEG decoding, but also facilitates the development of portable EEG devices.

4.2 Action Classification

4.2.1 Experiments Settings

In this study, an action classification model based on CNN is trained and evaluated using EMG data. The model is trained using 500 epochs and a batch size of 10, with



e) 4–6 s f) 6–8 s

g) 4–6 s h) 6–8 s

Figure 6. The 4-stage channel connectivity matrix and its corresponding connection diagram (a and c, b and d, e and g, f and h)

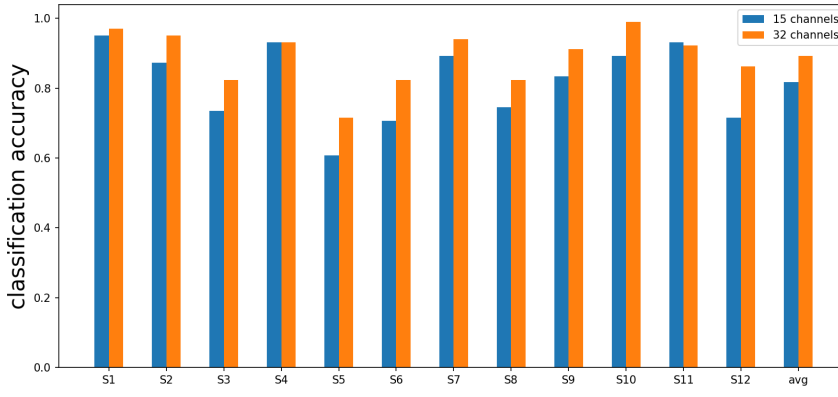


Figure 7. The accuracy rate: 15 channels vs 32 channels

the Adam optimizer and a learning rate of 1e−3. The loss function is defined as the sum of cross entropy between the predicted and actual labels. During evaluation, average accuracy of the test data is used as a metric to assess performance of the model.

4.2.2 Experiments Results

The present study aimed to decode EMG signals for the purpose of identifying different weights of lifted objects (i.e., 150 g, 300 g, 600 g). Effectiveness of the proposed model is evaluated using classification accuracy for these three types of data. Additionally, the proposed model is compared to a benchmark model, lightgbm [33], with the results shown in Figure 8.

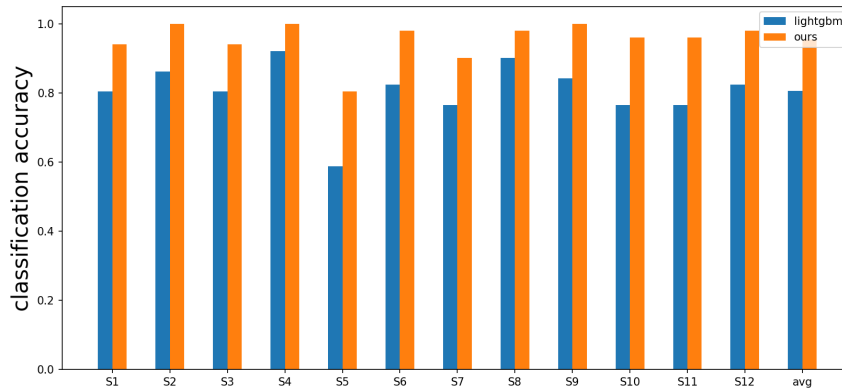


Figure 8. EMG: Classification accuracy for lifting different weights

Figure 8 illustrates the classification accuracy results for the proposed model, which are found to be excellent, with the exception of subject 5, for whom the accuracy is lower than 0.9. This may be attributable to poor signal quality from that subject. Furthermore, compared to the baseline model lightgbm, the proposed model demonstrated better performance across all tests. Specifically, the average accuracy of the proposed model (0.954) is about 0.15 higher than that of the baseline model (0.806), indicating its effectiveness.

To examine the generality of our model, we conduct additional experiments on different task formats. In a given trial (0–8 s), subjects performed a series of actions, including reaching out (2–4 s), lifting the object (4–6 s), and putting it down (6–8 s). By identifying these three actions, our model not only demonstrates its versatility but also provides opportunities for rehabilitation training. As shown in Figure 9, we compare the accuracy of the proposed model to that of the benchmark model, lightgbm. Overall, the classification accuracy of the proposed model is excellent, achieving an average accuracy of 0.937, which is 0.04 higher than that of the

benchmark model (0.897). These results further underscore the effectiveness and generality of our proposed model.

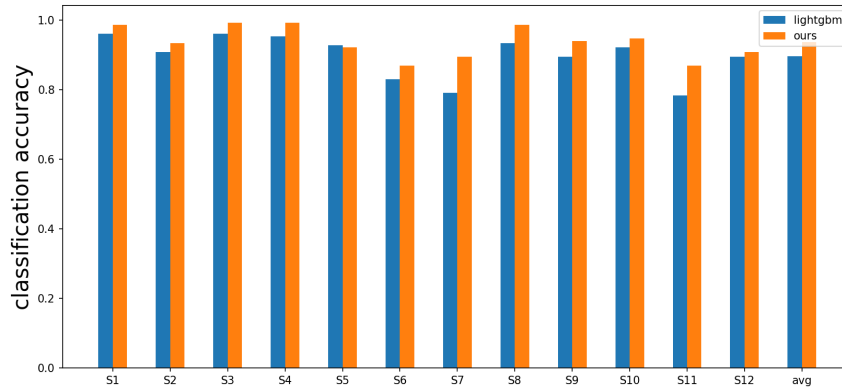


Figure 9. EMG: Classification accuracy of different actions

5 CONCLUSION AND FUTURE WORK

This study proposes a novel framework for hand rehabilitation training using EEG and EMG signals. EEG is used to detect movement intention, while EMG is utilized to recognize specific hand gestures. To decode EEG signals, we employ GCN-LSTM, which achieved an average classification accuracy of 0.892 surpassing the benchmark classifier (0.742) and demonstrating the effectiveness of our model. Additionally, we analyze channel connectivity to explain the interpretability of the model, finding that using a subset of highly connected channels (15 channels) resulted in only a 0.07 decrease in accuracy when the amount of data is halved, which indicates the potential for simplifying the number of EEG channels needed. Using a CNN, EMG signals are decoded to recognize different hand movements in two different tasks, with the proposed model achieving an average accuracy of 0.954 and 0.937, respectively, which outperformed the benchmark model lightgbm. These results highlight the effectiveness and generalizability of our proposed model for hand rehabilitation training. We can apply the framework proposed in this study to the hybrid BCI system, combined with the hardware equipment of the BCI, so as to realize the patient's hand rehabilitation training. Specifically, the user's movement intention is identified through EEG decoding, and EMG decoding is used to realize specific hand movements or fine power control, and then external devices such as mechanical gloves are used to assist the subject's movement, and the system gives the subject certain feedback. Through this series of processes, the patient's neural circuit is rebuilt to achieve rehabilitation training. In addition, future work in this area should focus on the following aspects:

- In the aspect of motion intention recognition based on EEG, by improving the model, effective monitoring can be carried out while reducing the number of data sampling points, thereby reducing the response time of the BCI system and improving usability.
- Further analyzing channel connectivity in EEG to improve interpretability and identify channels that are closely related to different actions. This can help select appropriate channels for specific tasks, thereby aiding in the development of portable BCI devices.
- Furthermore, the connectivity between EEG and EMG can be explored to discuss the mechanisms behind the operation of hybrid BCI systems.

6 FUNDING

This work is supported by the Shanghai Municipal Science and Technology Major Project (2021SHZDZX0100) and the Fundamental Research Funds for the Central Universities, and the Science and Technology Innovation Action Plan of the Shanghai Science and Technology Commission (19441908000).

REFERENCES

[1] Yue, Z.—Zhang, X.—Wang, J.: Hand Rehabilitation Robotics on Poststroke Motor Recovery. Behavioural Neurology, Vol. 2017, 2017, Art. No. 3908135, doi: 10.1155/2017/3908135.

[2] Nichols-Larsen, D. S.—Clark, P. C.—Zeringue, A.—Greenspan, A.—Blanton, S.: Factors Influencing Stroke Survivors' Quality of Life During Subacute Recovery. Stroke, Vol. 36, 2005, No. 7, pp. 1480–1484, doi: 10.1161/01.STR.0000170706.13595.4f.

[3] Jeannerod, M.: Mental Imagery in the Motor Context. Neuropsychologia, Vol. 33, 1995, No. 11, pp. 1419–1432, doi: 10.1016/0028-3932(95)00073-C.

[4] Müller-Putz, G. R.—Scherer, R.—Pfurtscheller, G.—Rupp, R.: EEG-Based Neuroprosthesis Control: A Step Towards Clinical Practice. Neuroscience Letters, Vol. 382, 2005, No. 1-2, pp. 169–174, doi: 10.1016/j.neulet.2005.03.021.

[5] Zhao, Q. B.—Zhang, L. Q.—Cichocki, A.: EEG-Based Asynchronous BCI Control of a Car in 3D Virtual Reality Environments. Chinese Science Bulletin, Vol. 54, 2009, No. 1, pp. 78–87, doi: 10.1007/s11434-008-0547-3.

[6] Meng, J.—Zhang, S.—Bekyo, A.—Olsoe, J.—Baxter, B.—He, B.: Noninvasive Electroencephalogram Based Control of a Robotic Arm for Reach and Grasp Tasks. Scientific Reports, Vol. 6, 2016, No. 1, Art. No. 38565, doi: 10.1038/srep38565.

[7] Foong, R.—Ang, K. K.—Quek, C.—Guan, C.—Phua, K. S.—Kuah, C. W. K.—Deshmukh, V. A.—Yam, L. H. L.—Rajeswaran, D. K.—Tang, N.—Chew, E.—Chua, K. S. G.: Assessment of the Efficacy of EEG-Based MI-BCI with Visual Feedback and EEG Correlates of Mental Fatigue for Upper-Limb

Stroke Rehabilitation. IEEE Transactions on Biomedical Engineering, Vol. 67, 2020, No. 3, pp. 786–795, doi: 10.1109/TBME.2019.2921198.

[8] Murphy, S. M.: Imagery Interventions in Sport. Medicine and Science in Sports and Exercise, Vol. 26, 1994, No. 4, pp. 486–494, doi: 10.1249/00005768-199404000-00014.

[9] Sharma, N.—Pomeroy, V. M.—Baron, J. C.: Motor Imagery: A Backdoor to the Motor System After Stroke? Stroke, Vol. 37, 2006, No. 7, pp. 1941–1952, doi: 10.1161/01.STR.0000226902.43357.fc.

[10] Kraeutner, S.—Gionfriddo, A.—Bardouille, T.—Boe, S.: Motor Imagery-Based Brain Activity Parallels That of Motor Execution: Evidence from Magnetic Source Imaging of Cortical Oscillations. Brain Research, Vol. 1588, 2014, pp. 81–91, doi: 10.1016/j.brainres.2014.09.001.

[11] Johnson, S. H.: Imagining the Impossible: Intact Motor Representations in Hemiplegics. NeuroReport, Vol. 11, 2000, No. 4, pp. 729–732, doi: 10.1097/00001756-200003200-00015.

[12] Babiloni, C.—Pizzella, V.—Del Gratta, C.—Ferretti, A.—Romani, G. L.: Chapter 5. Fundamentals of Electroencefalography, Magnetoencefalography, and Functional Magnetic Resonance Imaging. International Review of Neurobiology, Vol. 86, 2009, pp. 67–80, doi: 10.1016/s0074-7742(09)86005-4.

[13] Kipf, T. N.—Welling, M.: Semi-Supervised Classification with Graph Convolutional Networks. CoRR, 2016, doi: 10.48550/arXiv.1609.02907.

[14] Song, T.—Zheng, W.—Song, P.—Cui, Z.: EEG Emotion Recognition Using Dynamical Graph Convolutional Neural Networks. IEEE Transactions on Affective Computing, Vol. 11, 2020, No. 3, pp. 532–541, doi: 10.1109/TAFFC.2018.2817622.

[15] Zhang, T.—Wang, X.—Xu, X.—Chen, C. L. P.: GCB-Net: Graph Convolutional Broad Network and Its Application in Emotion Recognition. IEEE Transactions on Affective Computing, Vol. 13, 2019, No. 1, pp. 379–388, doi: 10.1109/TAFFC.2019.2937768.

[16] Manessi, F.—Rozza, A.—Manzo, M.: Dynamic Graph Convolutional Networks. Pattern Recognition, Vol. 97, 2020, Art. No. 107000, doi: 10.1016/j.patcog.2019.107000.

[17] Sun, M.—Cui, W.—Yu, S.—Han, H.—Hu, B.—Li, Y.: A Dual-Branch Dynamic Graph Convolution Based Adaptive TransFormer Feature Fusion Network for EEG Emotion Recognition. IEEE Transactions on Affective Computing, Vol. 13, 2022, No. 4, pp. 2218–2228, doi: 10.1109/TAFFC.2022.3199075.

[18] Asadzadeh, S.—Yousefi Rezaii, T.—Beheshti, S.—Meshgini, S.: Accurate Emotion Recognition Using Bayesian Model Based EEG Sources as Dynamic Graph Convolutional Neural Network Nodes. Scientific Reports, Vol. 12, 2022, No. 1, Art. No. 10282, doi: 10.1038/s41598-022-14217-7.

[19] Ye, M.—Chen, C. L. P.—Zhang, T.: Hierarchical Dynamic Graph Convolutional Network with Interpretability for EEG-Based Emotion Recognition. IEEE Transactions on Neural Networks and Learning Systems, 2022, doi: 10.1109/TNNLS.2022.3225855.

[20] Tavakoli, M.—Benussi, C.—Lopes, P. A.—Osorio, L. B.—de Almeida, A. T.: Robust Hand Gesture Recognition with a Double Channel

Surface EMG Wearable Armband and SVM Classifier. Biomedical Signal Processing and Control, Vol. 46, 2018, pp. 121–130, doi: 10.1016/j.bspc.2018.07.010.

[21] Chen, L.—Fu, J.—Wu, Y.—Li, H.—Zheng, B.: Hand Gesture Recognition Using Compact CNN via Surface Electromyography Signals. Sensors, Vol. 20, 2020, No. 3, Art. No. 672, doi: 10.3390/s20030672.

[22] Côté-Allard, U.—Fall, C. L.—Drouin, A.—Campeau-Lecours, A.—Gosselin, C.—Glette, K.—Laviolette, F.—Gosselin, B.: Deep Learning for Electromyographic Hand Gesture Signal Classification Using Transfer Learning. IEEE Transactions on Neural Systems and Rehabilitation Engineering, Vol. 27, 2019, No. 4, pp. 760–771, doi: 10.1109/TNSRE.2019.2896269.

[23] Toro-Ossaba, A.—Jaramillo-Tigreros, J.—Tejada, J. C.—Peña, A.—López-González, A.—Castanho, R. A.: LSTM Recurrent Neural Network for Hand Gesture Recognition Using EMG Signals. Applied Sciences, Vol. 12, 2022, No. 19, Art. No. 9700, doi: 10.3390/app12199700.

[24] Leeb, R.—Sagha, H.—Chavarriaga, R.—del R. Millán, J.: A Hybrid Brain-Computer Interface Based on the Fusion of Electroencephalographic and Electromyographic Activities. Journal of Neural Engineering, Vol. 8, 2011, No. 2, Art. No. 025011, doi: 10.1088/1741-2560/8/2/025011.

[25] Lin, K.—Cinetto, A.—Wang, Y.—Chen, X.—Gao, S.—Gao, X.: An Online Hybrid BCI System Based on SSVEP and EMG. Journal of Neural Engineering, Vol. 13, 2016, No. 2, Art. No. 026020, doi: 10.1088/1741-2560/13/2/026020.

[26] Sarasola-Sanz, A.—Irastorza-Landa, N.—López-Larraz, E.—Bibián, C.—Helmhold, F.—Broetz, D.—Birbaumer, N.—Ramos-Murguialday, A.: A Hybrid Brain-Machine Interface Based on EEG and EMG Activity for the Motor Rehabilitation of Stroke Patients. 2017 International Conference on Rehabilitation Robotics (ICORR), 2017, pp. 895–900, doi: 10.1109/ICORR.2017.8009362.

[27] Tun, N. N.—Sanuki, F.—Iramina, K.: Electroencephalogram-Electromyogram Functional Coupling and Delay Time Change Based on Motor Task Performance. Sensors, Vol. 21, 2021, No. 13, Art. No. 4380, doi: 10.3390/s21134380.

[28] Tun, N. N.—Sanuki, F.—Iramina, K.: EEG-EMG Correlation Analysis with Linear and Nonlinear Coupling Methods Across Four Motor Tasks. 2021 43rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2021, pp. 783–786, doi: 10.1109/EMBC46164.2021.9629969.

[29] Soundirarajan, M.—Krejcar, O.—Namazi, H.: Evaluation of the Coupling Between the Brain and Facial Muscles Reactions to Moving Visual Stimuli. Fluctuation and Noise Letters, Vol. 20, 2021, No. 5, Art. No. 2150042, doi: 10.1142/S0219477521500425.

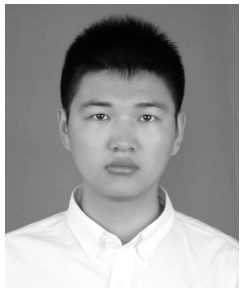
[30] Luciw, M. D.—Jarocka, E.—Edin, B. B.: Multi-Channel EEG Recordings During 3,936 Grasp and Lift Trials with Varying Weight and Friction. Scientific Data, Vol. 1, 2014, No. 1, Art. No. 140047, doi: 10.1038/sdata.2014.47.

[31] Afshani, F.—Shalbaf, A.—Shalbaf, R.—Sleigh, J.: Frontal-Temporal Functional Connectivity of EEG Signal by Standardized Permutation Mutual Information During Anesthesia. Cognitive Neurodynamics, Vol. 13, 2019, No. 6, pp. 531–540, doi:

10.1007/s11571-019-09553-w.

[32] Ang, K. K.—Chin, Z. Y.—Zhang, H.—Guan, C.: Filter Bank Common Spatial Pattern (FBCSP) in Brain-Computer Interface. 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), 2008, pp. 2390–2397, doi: 10.1109/IJCNN.2008.4634130.

[33] Ye, Y.—Liu, C.—Zemiti, N.—Yang, C.: Optimal Feature Selection for EMG-Based Finger Force Estimation Using LightGBM Model. 2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), 2019, pp. 1–7, doi: 10.1109/RO-MAN46459.2019.8956453.



Lubo Fu received his B.Sc. degree in information security from the Tongji University in 2021. Now, he is a graduate student in the Department of Computer Science and Technology, Tongji University. His research interests include machine learning and brain-computer interface.

Haoyang Li received his B.Sc. degree in software engineering from the Nantong University in 2020. Now, he is a graduate student in the Department of Computer Science and Technology, Tongji University. His research interests include machine learning and brain-computer interface.

Hongfei Ji is an Associate Professor with the Department of Computer Science and Technology, Tongji University, Shanghai, China. He received the Ph.D. degree in software engineering from the Tongji University, Shanghai, China, in 2015. His research interests include machine learning, pattern recognition, brain computer interface, and cognitive neuroscience.

Jie Li is an Associate Professor with the Department of Computer Science and Technology, Tongji University, Shanghai, China. She received the Ph.D. degree in computer science from the Shanghai Jiao Tong University, Shanghai, China, in 2010. Her research interests include machine learning, brain computer interface, brain like computation, and cognitive neuroscience.

Computing and Informatics, Vol. 42, 2023, 762–780, doi: 10.31577/cai_2023_3_762

OPTIMIZATION OF COLUMNAR NOSQL DATA WAREHOUSE MODEL WITH CLARANS CLUSTERING ALGORITHM

Nassima Soussi

LIPIM, National School of Applied Sciences
Sultan Moulay Slimane University
Khouribga, Morocco
e-mail: nassima.soussi@gmail.com

Abstract. In order to perfectly meet the needs of business leaders, decision-makers have resorted to the integration of external sources (such as Linked Open Data) in the decision-making system in order to enrich their existing data warehouses with new concepts contributing to bring added value to their organizations, enhance its productivity and retain its customers. However, the traditional data warehouse environment is not suitable to support external Big Data. To deal with this new challenge, several researches are oriented towards the direct conversion of classical relational data warehouse to a columnar NoSQL data warehouse, whereas the existing advanced works based on clustering algorithms are very limited and have several shortcomings. In this context, our paper proposes a new solution that conceives an optimized columnar data warehouse based on CLARANS clustering algorithm that has proven its effectiveness in generating optimal column families. Experimental results improve the validity of our system by performing a detailed comparative study between the existing advanced approaches and our proposed optimized method.

Keywords: Big Data, columnar NoSQL data warehouse, linked open data, clustering algorithms, Clarans

1 INTRODUCTION

Since many decades, data warehouse system has been a very important place in data analytics solutions thanks to its ability to effectively manage one of the major capital of any organization: the Data. However, the arrival of big data and the need

for analysis continuously this voluminous mass in perpetual increase has impacted its ecosystem. This situation has led some analysts to anticipate the disappearance of DW in favor of Big Data systems, whereas the majority of data warehousing communities are campaigning for its extension [1] in order to align their decisional systems to Big Data requirements [2]. This alignment is materialized principally through the integration of Big Data pillars summarized in 5 Vs (Volume, Variety, Velocity, Value and Veracity) in the different phases of decision-making process.

In traditional data warehouses, a large number of requirements are not fully met by internal sources. This situation penalizes companies looking for real added value and contradicts the initial objective of decision-making systems. Hence, in order to help managers to make the right decisions, companies owning data warehousing technology have to enrich their existing data warehouses with new concepts by integrating external Big Data (related to their activities) in their decision-making system. However, traditional DW based on relational database is not suitable to support external big data characterized by its high volume and data format heterogeneity which requires a schemaless distributed system able to cohabit internal and external data efficiently contributing to the renaissance of this vital ecosystem.

To deal with the previous challenges, several researches that have been proposed in the literature are oriented towards the direct conversion of classical relational data warehouse to a columnar NoSQL data warehouse (CN-DW), whereas the few advanced works based on clustering algorithms are very limited and have several shortcomings. Thus, the issue of designing an optimized CN-DW still arises. In this paper, we revisit existing CN-DW models by treating its main limitations in our new solution that conceives an optimized model based on clustering algorithm [3] with an optimal column family's number. In addition, our system takes into consideration all relational data warehouse (Rel-DW) attributes to design a complete targeted model able to meet perfectly the new needs of business leaders.

The remainder of this paper is organized as follows: Section 2 presents the most recent related works in the current topic and discusses their main limitations. Section 3 highlights the main concepts used in the proposed solution, such as the HBase NoSQL database and Clarans clustering algorithms. Section 4 presents the functional architecture of our optimized CN-DW with a set of detailed algorithms for each phase. Section 5 improves the performance of our system with a real and practical comparison with the most recent existing approaches. Section 6 presents an analysis and discussion. Finally, Section 7 concludes our work and suggests some future extensions of this topic.

2 RELATED WORKS

In order to perfectly meet the needs of business leaders in the big data era, several researches have been made to ensure the enrichment of the decisional systems with different types of big data such as semantic web [4], social data [5, 6], NoSQL data [7], data lakes [8] and LOD [9, 10, 11]. However, neither of these approaches

has implemented a decision-making system based on a big data warehouse, which is ready at all times to support the increased integration of big data so as to constantly satisfy the functional exigencies of decision-makers. To deal with this challenge, and to ensure a suitable environment for external big data, numerous works have been design a big data warehouse based on column oriented NoSQL database (CN-DW) from a classical DW (Rel-DW) trying to respect the maximum of specifications and peculiarities of each system. The existing works can be classified into three main categories:

Naive approaches: These approaches such as [12, 13, 14] and [15] are called naives since they propose a direct conversion method based on a set of mapping rules defined by matching the basic characteristics of each model. However, all these works neglected the consideration of any optimization operation; hence they have an imbalance on the number of attributes in column families. In addition, they did not control the number of generated column families.

Advanced approaches: In order to solve the shortcoming and weaknesses of naive approaches, several works have been made to enhance the existing CN-DW schema conception and group attributes in column families more efficiently. In [16], the authors decided to classify Rel-DW attributes into two column families in CN-DW schema: the first one gathers the most interrogated attributes, whereas the second one groups the remaining attributes. In addition, paper [17] presents a new method called NoSE (NoSQL Schema Evaluator) considered as a cost-based method aiming to recommend an optimized storage schemas for NoSQL column oriented database. However, the effectiveness of this system is limited to queries manipulating a small number of attributes. But all these approaches do not consider clustering techniques in their CN-DW conception methods to optimize the grouping of column families in order to improve the query processing performance and response times to complex queries.

Optimized approaches: In order to design the most optimized data model of CN-DW column families, the first work established in this direction is presented in [18] that propose a new method for grouping data carefully (from fact and dimensions of Rel-DW) in one CN-DW table using k-means as a clustering algorithm. In fact, the authors use this technique in order to group in the same column family the frequently used attributes based on a set of decisional queries. Likewise, the paper [19] addresses the same issue, with a normalized technique which consists to create deferent tables in CN-DW instead of one table. To do that, they grouped analogous queries in classes using k-medoid algorithm, end then they used the PSO algorithm (Particle Swarm Optimization algorithm based on meta-heuristics) to gather column families attributes according to each similar class of queries. The experimental results obtained by adopting clustering algorithms in CN-DW design prove the enhancement of decisional queries performance. Although, these optimized researches share some similarities with our solution presented in this paper, however they neglected two important points:

- The both clustering algorithms adopted above (k-means and k-medoid [20]) require the specification of clusters number without proposing any improvement of this initial configuration.
- They consider just the attributes used in decisional queries to conceive the distributed data warehouse. In this case, the designed CN-DW will be unable to meet new needs requiring the use of excluded attributes.

This comparative study showed that, despite these numerous solutions proposed in the literature (discussed previously), the issue of designing an optimized CN-DW still arises. To deal with the previous shortcomings, our current work provides a new approach aiming to design a CN-DW with an optimal clusters number with CLARANS algorithm. In addition, our system take into consideration all Rel-DW attributes to design a complete targeted model able to meet perfectly the new needs of business leaders.

Table 1 presents a technical comparison of the most recent approaches aiming to design CN-DW from Rel-DW. To realize this comparison, we considered the following main characteristics:

Conversion type: indicates the type of CN-DW design: (a) direct methods are based on a set of mapping rules defined by matching the basic characteristics of each model. (b) advanced methods are more developed than the first ones; for exemple they gather the most interrogated attributes in one columns, and the remaining attributes in the second one. (c) optimized methods design the most optimized data model of CN-DW.

Type of optimization technique used by optimized approaches.

Name of optimization technique

Complete model: in order to have a complete CN-DW that can meet future needs, it should contains all the Rel-DW attributes even if they are not used by current analytical questions.

Predefined CF number: it refers to the number of CN-DW column families: is it predefined randomly or generated and optimized by clustering algorithms.

Type of CN-DW: the NoSQL system used to implement the targeted CN-DW.

3 BACKGROUND

In this section, we introduce the main concepts used by our solution presented in Figure 1, that describes the proposed optimal CN-DW implemented in HBase as a column oriented NoSQL database and based on Clarans clustering algorithm to design the best regroupement of columns families.

	Conversion Type	Type of Optimization Techniques	Name of Optimization Technique	Complete Model	Predefined CF Number	Type of CN-DW
[12]	Direct			Yes	Yes	HBase
[13]	Direct			Yes	Yes	Hbase
[14]	Direct			Yes	Yes	HBase
[15]	Direct			Yes	Yes	Cassandra
[16]	Advanced			Yes	Yes	HBase
[17]	Advanced			Yes	Yes	Cassandra
[19]	Optimized	Clustering & meta-heuristic algorithms	K-medoid + PSO	No	Yes	HBase
[18]	Optimized	Clustering algorithms	K-means	No	Yes	HBase

Table 1. Technical comparison of CN-DW design approaches

3.1 HBase

HBase [21] is a Column Oriented Database that looks remarkably like a relational database, but the concept is entirely different. It is dedicated to accommodating many columns (up to several million) for each line. It is characterized by a variable number of columns that can change from one row to another (we can consider that a column exists if it contains a value). This type of NoSQL database offers a high scalability in data storage and flexible schema due to the number of columns that can change from one row to another. The model of column-oriented database is composed of a set of Tables; each table contains a set of rows. Each row can be represented as $Ri = (Idi, (CFi1, CFi2, \ldots, CFim))$ with $i \in [1, n]$, $j \in [1, m]$, Idi is a row id and $CFij$ is a column family of the row Ri. Each column family can contain numerous columns that have the same categories of attributes which also called Column Qualifier. In this work we have chosen HBase to implement the CN-DW in order to have a flexible schema ready to receive heterogeneous external data easily and to improve the execution time of decision queries. In addition, we can operate on HBase tables with Spark and MapReduce for parallel processing of big data.

3.2 Clustering Algorithms

Automatic classification or clustering is an important step in the process of Knowledge Extraction from Data. It aims to develop an optimal partitioning by grouping data into classes that share similar characteristics where the data is generally represented by measurement vectors or points in a multidimensional space. Intuitively, vectors belonging to a valid cluster are more similar to each other than a vector

belonging to a differ group. In other words, the goal of these methods is to identify groups from an unlabeled set of data vectors that share semantic similarities. This allows the user to construct a cognitive model, thus aiding the detection of the inherent structure of a data set.

There are four main families of clustering algorithms: Hierarchical algorithms, Algorithms by partition, Algorithms based on density, Classification based on quantification by grid. In this paper, the most suitable type for our problem is the data partitioning algorithms in order to design the best grouping of column families for the CN-DW. They consist in dividing directly a set of data (the attributes of Rel-DW) into k classes (or families of columns in our case) such that each class (or FC) must contain at least one attribute and each attribute must belong to a unique class unlike the so-called fuzzy classification which does not impose this condition. Among the famous algorithms of this type [22], we quote: K-means, K-medoids, Partition Around Medoid (PAM), Clustering LARge Applications (CLARA) et Clustering large applications based upon randomized search (CLARANS).

The general algorithm of a partition classification follows the steps below:

1. Determine the number of clusters.
2. Initialize cluster centers.
3. Partition the dataset.
4. Calculate cluster centers (make an update).
5. If the partitioning is unchanged (or the algorithm has converged), stop; otherwise go to step 3.

A problem that accompanies the use of a partition classification algorithm is the choice of the desired output classes number. Partition clustering techniques generally produce clusters by optimizing an objective function defined locally (on a subset of data vectors) or globally (defined on all vectors) which translates that the objects must be similar within a same class, and dissimilar from one class to another. For a classification in k classes, these algorithms generate an initial partition, and then search to improve it by reassigning individuals from one class to another, which allows the possibility that a poor initial partition could be corrected later.

In our context of designing an optimized CN-DW, we use Clarans (as a partition algorithm) in order to propose the best grouping of Rel-DW attributes in the targeted CN-DW columns families. Indeed, it makes a stochastic search based on different parameters allowing to limit the number of iterations of the method, as well as on random sampling. Given k the number of clusters sought, a data partitioning consists of a set of k medoids, to which are associated the set of objects according to their proximity to these medoids. The main steps of the method are as follows:

1. select a representative sample of data;
2. iterate a fixed number of times;

    - choose a random solution: a set of k medoid;

- iterate a fixed number of times:
  - choose a neighboring solution of the current one by a random modification of one of the solution's medoid;
  - keep the neighbor as the new current solution if the global inertia of the partition is less than the previous solution inertia;
- store the local optimal solution found

3. return the best of the local optimal solutions found.

CLARANS allows to extract classes of better quality compared to the PAM and CLARA methods; however this method is sensitive to the chosen parameters and has a complexity of the order $O(k.n^2)$.

4 PROPOSED APPROACH

In this section, we describe our novel contribution illustrated via the functional architecture presented in Figure 1. This architecture is composed of five principal layers:

Rel-DW layer: contains the data source of our system as a relational DW (Rel-DW) with its metadata and the most frequent decisional queries with their access frequencies.

Rel-DW to CN-DW layer: presents the main phase in our proposed solution that operates on the previous elements in order to group each set of attributes used together (in the selected input decisional queries) in the same column family to design an optimized targeted CN-DW, with Clarans clustering algorithm, that take into consideration all Rel-DW attributes in order to design a complete targeted model.

CN-DW layer: the logical optimized schema provided by the second layer is used to implements the CN-DW on HBase [23], considered as the most used NoSQL column oriented DB. This new system is fed from input source (Rel-DW) by respecting its logical schema.

Decision Maker & CN-DW Enrichment Layers: these layers correspond to the on-demand Big ETL that feed the CN-DW from external big data sources (Linked Open Data, Data Lakes, social media, semantic data, etc.) only when the decision makers need to enrich some answers of decisional queries judged as unsatisfactory to the needs of managers. In fact, this phase requires the adaptation of internal decision queries to external system schema in order to be able to extract perfectly the required data fragments. The external results are exploited in two ways:

1. they are integrated into CN-DW for future analyzes more fruitful;
2. they are merged with the results of internal decision query in order to be visualized by end users for more efficient managerial decisions.

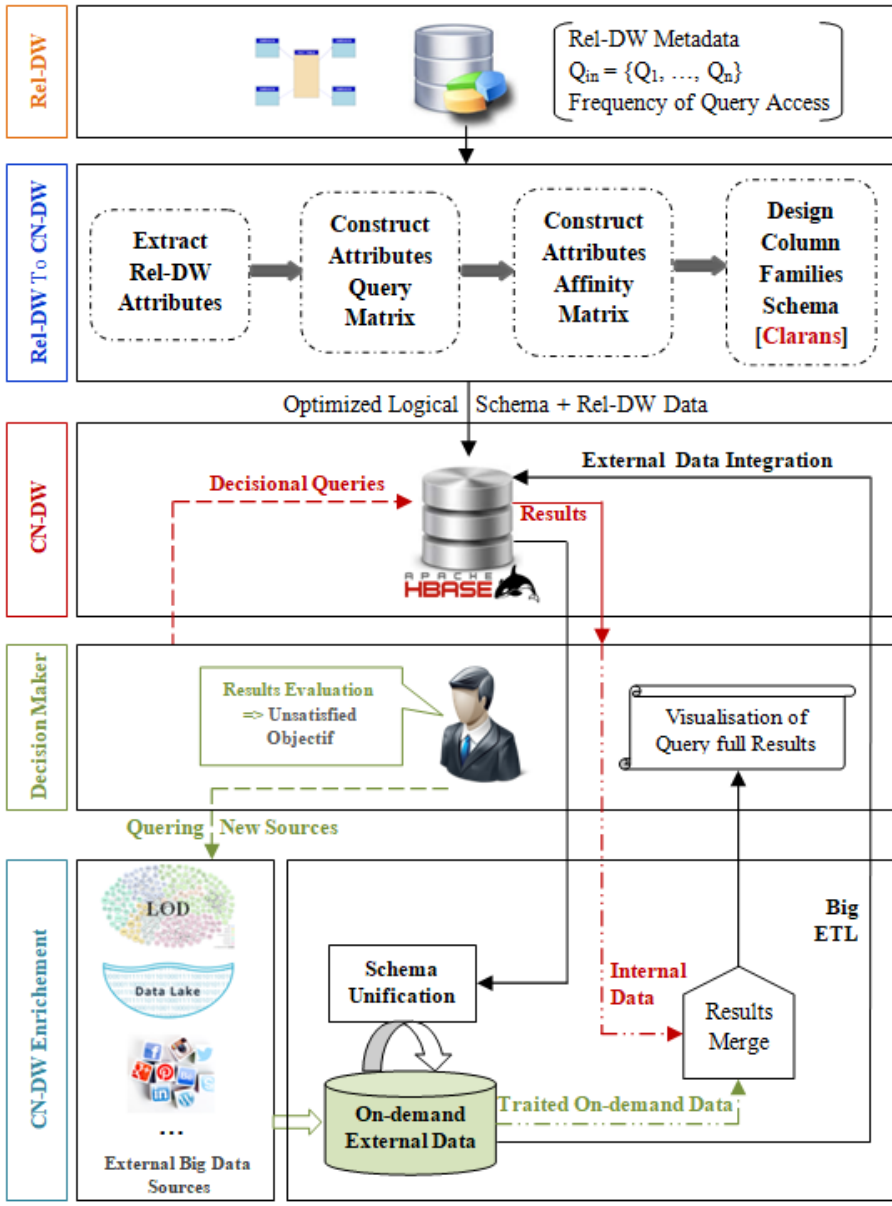


Figure 1. Global Architecture of our approach

In this paper, we will focus specially on the second layer designing an optimized CN-DW with Clarans clustering algorithm as presented in Algorithm-1. It operates on relational DW metadata (Rel-DW), list of most frequent decisional queries (Qdecisional), Frequency of Query Access Matrix by sites (FQAM), initial number of cluster defining column families (k), maximal number of neighbors (max_neighbor) and the iterations number of CLARANS clustering algorithm to resolve the problem (iterationsNbr) in order to design an optimized model for the targeted DW based on NoSQL column-oriented DB (CN-DW) that we represent as a big column NoSQL table (TCN-DW). In fact, our approach processes according to five major steps as described in Algorithm-1:

- Step 1: Extract the set of attributes
- Step 2: Construct Attributes Query Matrix (AQM)
- Step 3: Construct Attributes Affinity Matrix (AAM)
- Step 4: Design column family schemas (with Clarans algorithm)
- Step 5: Create final column-oriented NoSQL table (CN-DW)

Algorithm-1 : Rel-DW to CN-DW
Inputs : Rel-DW-metadata, List $Q_{decisional}$, FQAM, k, max_neighbor, iterationsNbr
Output : $T_{CN\text{-}DW}$
Begin
List L_{attr}, AQM, AAM
List<table[]> CFs<Attributes>
1. $L_{attr} \leftarrow$ ExtractAttributes(Rel-DW, List $Q_{decisional}$)
2. AQM $\leftarrow$ ConstructAttributesQueryMatrix(L_{attr}, $Q_{decisional}$)
3. nbrQ $\leftarrow$ $Q_{decisional}$.lenght() //number of queries
4. nbrAtt $\leftarrow$A.lenght() //number of attributes
5. AAM $\leftarrow$ ConstructAttributesAffinityMatrix(AQM, FQAM, nbrQ, nbrAtt)
6. CFs $\leftarrow$ Clarans(AAM, k, max_neighbor, iterationsNbr)
7. $T_{CN\text{-}DW} \leftarrow$ CreateOrientedColumnTable(table_name, CFs)
8. **Return** $T_{CN\text{-}DW}$
End Algorithm

Figure 2. Algorithm-1

4.1 Extract Rel-DW Attributes

We aim in this step to prepare the main object in our major process to design column families in the targeted DW. In fact, and as illustrated in Algorithm-2, we operate on Rel-DW metadata to access its dimensions and facts in order to return all Rel-DW attributes that will be used by the next step.

Algorithm-2 : Extracting Rel-DW Attributes
Input : Rel-DW-metadata
Output : $\text{Liste}_{\text{Attr}}$
Begin
1. M ← Parse(Rel-DW-metadata)
2. D ← ExtractDimensions(M)
3. F ← ExtractFacts(M)
4. **Foreach** Table **in** F or D **do**
5. A ← ExtractAttributes(Table)
6. **Foreach** elt **in** A **do**
7. $\text{Liste}_{\text{Attr}}$.Add(A)
8. **End For**
9. **End For**
10. **Return** $\text{Liste}_{\text{Attr}}$

End Algorithm

Figure 3. Algorithm-2

4.2 Construct Attributes Query Matrix

In order to design CN-DW columns by grouping a set of attributes used together (in decisional queries) in the same column family, we have to conceive firstly the Attributes Query Matrix (AQM) as illustrated in Algorithm-3. It operates on the previous list of Rel-DW attributes $ListeAttr\{a_1, \ldots, a_n\}$ (generated by Algorithm-2) and a list of the most frequent decisional queries $QOLAP = \{q_1, \ldots, q_m\}$ presented as inputs in order to construct the AQM dimensioned by $n * m$ where n is the number of $QOLAP$ and m is the number of attributes. In fact, we go throught the previous lists to check the membership of each attribute to the set of queries: if an attribute a_i appears in q_j then the $AQM[i, j]$ is equal to 1 else $AQM[i, j]$ is equal to 0.

4.3 Construct Attributes Affinity Matrix

AAM is a symmetric matrix ($n \times n$ where n is the number of attributes) that indicates how the attributes are closely related. Each element of the AAM matrix is defined by an affinity value which measures the strength of an imaginary link between two attributes based on the fact that these attributes are used together by the query. The affinity value between two attributes a_i and a_j represents the number of times where two attributes are accessed together on all sites, it is defined as follows:

$$Aff(a_i, a_j) = \Sigma_{\text{All queries that access } a_i \text{ and } a_j} \textbf{Query access}, \tag{1}$$

where

$$\textbf{Query access} = \Sigma_{\text{For all sites}} \textbf{Access frequency of query}. \tag{2}$$

```
Algorithm-3: Construct Attributes Query Matrix
Inputs: List Liste_Attr, List Q_Decisional
Output: List AQM
Begin
1.  For i ← 1 to A. lenght() do
2.    For j ← to Q.lenght() do
3.      If(Liste_Attr[i].AppearsIn(Q[j]) = True) then
4.        AQM[i, j] ← 1
5.      Else
6.        AQM[i, j] ← 0
7.      End if
8.    End For
9.  End For
10. Return AQM
End Algorithm
```

Figure 4. Algorithm-3

The Algorithm-4 below describes the detailed conception steps of the AAM matrix. It takes as inputs AQM generated by Algorithm-3, Frequency of Query Access Matrix, queries number and the total number of attributes in Rel-DW in order to design Attributes Affinity Matrix (AAM) by implementing the previous equations.

4.4 Design Column Family Schemas with Clarans

In order to overcome the limitations of using k-means and k-medoid for designing a CN-DW in existing approaches as detailed in the previous section, we opted in our solution for Clarans algorithm. In fact, CLARANS (*Clustering large Application Based on Randomized Search*) is a partitioning method for clustering a large database [20]. It has proven its effectiveness in generating an optimal number of clusters unlike the first two algorithms keeping the same number initially proposed without any improvement. The Algorithm-5 describes the main conception steps adopted by Clarans to design an optimal grouping of CN-DW column families.

4.5 Create Column Oriented Table

After generating the schema of columns families (CFs) by Clarans clustering algorithm (Algorithm-5), the Algorithm-6 (as described below) operates on these CFs to create an HBase column-oriented table which will be ready to receive and group optimally the data from Rel-DW.

Algorithm-4 : Construct Attributes Affinity Matrix
Inputs : AQM, FQAM (Frequency of Query Access Matrix), nbrQ (nbr of queries), nbrAtt (total number of attributes)
Output : AAM (Attributes Affinity Matrix)
Begin
$FQAM_{reduced}$[nbrQ, 1] = Null
//Constructing reduced FQAM : Frequency of Query Access for all sites
1. **For** i ← 1 **to** nbrQ **do** *//FQAM rows*
2. **For** j ← **to** nbrSites **do** *//FQAM columns*
3. $FQAM_{reduced}$[i, 1] ← $FQAM_{reduced}$[i, 1] + FQAM[i, j]
4. **End For**
5. **End For**
6. *//Constructing AAM*
7. **For** i ← 1 **to** nbrAtt **do** *//AAM rows*
8. **For** j ← **to** nbrAtt **do** *//AAM columns*
9. **If** (i=j) **then**
10. **For** k ← 1 **to** nbrQ **do** *//browse the AQM matrix rows*
11. **If** (AQM[k, j] = 1) **Then**
12. AAM[i, j] ← AAM[i, j] + $FQAM_{reduced}$[k, 1]
13. **End If**
14. **End For**
15. **Else**
16. **For** k ← 1 **to** nbrQ **do** *//browse the AQM matrix rows*
17. **If** (AQM[k, i] = 1 **AND** AQM[k, j] = 1) **Then**
18. AAM[i, j] ← AAM[i, j] + $FQAM_{reduced}$[k, 1]
19. **End if**
20. **End For**
21. **End if**
22. **End For**
23. **End For**
24. **Return** AAM
End Algorithm

Figure 5. Algorithm-4

5 EXPERIMENTS RESULTS

In order to evaluate the performance of our system, we have implemented three different methods with python language: two already existing approaches using k-means and k-medoid clustering algorithm in addition to our method operating with Clarans, for designing an optimized CN-DW HBase system. We have operated on TPC-DS benchmark1 as a dataset. It is a relational data warehouse as a constellation schema, but in our context, we extracted a start schema composed of the fact table store_sales with its dimensions (Customer, Customer demographics, Customer address, Item, Time, Date, Household demographics, Promotion, Store). This start schema produces 176 attributes. To carry out our experiments, we set up two storage environments: the first one is relational with Intel-Core machine TMi5-4210U CPU@1.70 GHZ with 8 GB of RAM, and a 250 GB disk; it runs under the Windows 10 operating system with 64-bit. The second is a distributed NoSQL storage

```
Algorithm-5 : Design Columns Families schemas with Clarans
Inputs : AAM, k, max_neighbor, iterationsNbr
Output : medoidsOptimal
Begin
1.  OptimalCost ← 1000000
2.  medoidsOptimal ← Null
3.  i ← 1
4.  Do
5.  CurrentMedoids ← ConstructColumnFamilies(AAM, k)
6.  j ← 1
7.  Do
8.      NewMedoids ← RandomNeighborMedoids(CurrentMedoids)
9.      If (TotalCost_NewMedoids < TotalCost_CurrentMedoids) Then
10.         CurrentMedoids ← NewMedoids
11.     Else
12.         j ← j +1
13.     End If
14. While (j < max_neighbor)
15. If (TotalCost_CurrentMedoids < OptimalCost) Then
16.     OptimalCost ← TotalCost_CurrentMedoids
17.     medoidsOptimal ← CurrentMedoids
18. End if
19. i ← i +1
20. While (i < iterationsNbr)
21. Return medoidsOptimal
End Algorithm
```

Figure 6. Algorithm-5

environment, as a Cloudera virtual machine version 5.13.0 with 7 GB of RAM and 2 CPU.

In order to measure the effectiveness and the quality of each method, we have fixed some evaluation factors to do this:

1. the number of column families generated,
2. the cost of the attribute groups schema and
3. the execution time.

The number of column families generated: As presented in Table 2, our proposed approach (Clarans CN-DW) optimizes the NbrCF by generating a lower number than the initial one from a NbrCF equals to 8, unlike the other existing methods (k-means and k-medoid CN-DW) that design CN-DW based on the initial NbrCF without any optimization. We observe that for NbrCF between 160 and 175, our method generates an optimal schema with final NbrCF equals to 45 and a Square Error (defined in the sub-section bellow) equals to zero.

Algorithm-6 : Create Oriented Column Table
Inputs : String table_name, **List** CFs
Output : $T_{CN\text{-}DW}$
Begin
1. Instance ← DBconnection()
2. Instance.openDB()
3. Q_{select} ← **'create'** + table_name
4. **Foreach** elt **in** CFs **do**
5. Q_{select} ← Q_{select} + ',' + CFs
6. **End For**
7. **Return** $T_{CN\text{-}DW}$

End Algorithm

Figure 7. Algorithm-6

Initial NbrCF	2	8	10	25	30	40	160	168	170	172	174	175
Generated NbrCF	2	6	7	9	15	16	45	45	45	45	45	45

Table 2. Number of columns families generated by Clarans CN-DW method

The cost of the attributes groups schema: this metric is obtained using the Square Error (E2) as presented in [24]. In fact, the clustering method becomes more and more optimal as long as the E2 value approaches zero. The Square Error is calculated as follow:

$$E_S^2 = \sum_{i=1}^{k} \sum_{l=1}^{n} \left[(f_{q_l})^2 \times \alpha_i^{q_l} \left(1 - \frac{\alpha_i^{q_l}}{\beta_i} \right) \right] \tag{3}$$

with f_{q_l} is the access frequency of q_l query, αq_{l_i} is the number of attributes in CF_i accessed by q_l query, β_i is the number of attributes in column family CF_i.

In order to measure the quality of the attributes groups schema, we have calculated the Square Error of each method by varying the number of columns families (NbrCF). As presented in Table 3, the Square Error is equal to zero for K-means and K-medoid CN-DW methods from NbrCF = 45. However, the Square Error of Clarans CN-DW method is equal to zero from NbrCF = 160.

	K-Means	**K-Medoid**	**Clarans**
NbrCF	45	45	45
Square Error	0.0		

Table 3. Square Error for the three CN-DW methods

The execution time: the graph illustrated in Figure 2 describes the execution time variation for the three CN-DW methods (using K-means, K-medoid and Clarans clustering algorithm) by varying the number of columns families (NbrCF) between 2 and 45. In fact, we notice that the execution time of the

Clarans CN-DW method is very high compared to the other ones. However, as described in Figure 3, the execution time of the K-means CN-DW method is higher than K-medoid CN-DW one for $2 \leq \text{NbrCF} \leq 9$. On the other hand, for $10 \leq \text{NbrCF} \leq 45$ the execution time of K-medoid CN-DW method is very high compared to K-means one. Hence, we deduce the following arrangement of the three compared CN-DW methods: K-means < K-medoid < Clarans.

Table 4 presents a detailed execution time of generating CN-DW's CF number with K-means, K-medoid and Clanrans clustering algorithms.

		K-Means	**K-Medoid**	**Clarans**
Number of initial Columns families	2	0.062479	0.031249	13.790757
	10	0.12484	0.154433	8.250328
	25	0.289458	0.563852	23.487544
	30	0.285789	0.693265	17.856511
	40	0.364845	0.858601	23.44205
	45	0.409217	1.029528	26.946319

Table 4. Execution time of generating CN-DW's CF number with clustering algorithms

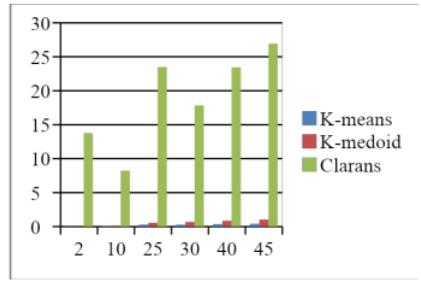


Figure 8. Execution time of generating CN-DW's CF number with K-means, K-medoid and Clarans

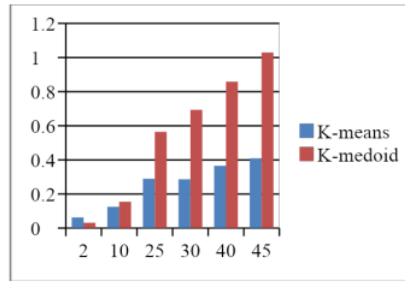


Figure 9. Comparison between K-means and K-medoid execution time

6 DISCUSSION

By analyzing the previous experiments results summarized in Table 5, we have noted that in the optimal case when the Square Error is equal to zero, the attributes not used by the input decisional queries are grouped in the same column family by our proposed method with Clarans, unlike the two existing methods which manipulate just the decisional queries attributes that conceive an incomplete CN-DW unsuitable for new needs. Although our approach has recorded a high execution time compared to existing methods, Table 2 proven its effectiveness in optimizing the number of column families unlike the two existing methods which preserves the same initial number and do not offer any improvement in this direction. In fact, our proposed approach always generates a number of columns families lower than the initial number especially in the case where this initial number is equal to (the number of attributes -1), our method returns an optimal schema in terms of execution time, number of clusters and Square Error. In addition, the execution time of our method depends on the number of iterations and the number of neighbors, if these parameters are small, the execution time is fast but the schema is not optimal, in the opposite case, the schema is optimal but the execution time is considerable.

	Initial CF Number	**Final CF Number**	**Execution Time**	**Square Error**
K-Means	45	45	0.405	0.0
K-Medoid	45	45	1.029	0.0
Clarans	175	45	20.480	0.0

Table 5. Summary of the main experiments results

7 CONCLUSION

In this contribution, we have proposed a new method to optimize designing columnar data warehouse using CLARANS clustering algorithm that generates an optimal grouping of column families. In addition, our system take into consideration all Rel-DW attributes to design a complete targeted model able to meet perfectly the new needs of business leaders. In order to improve the effectiveness and benefits of our system, we have elaborated numerous comparison tests (with existing works) based on TPC-DS[1] benchmark as a dataset. As future work, we intend to implement our CN-DW with the generated combination of columns families by CLARANS algorithm and execute the set of selected decisional queries on our optimized CN-DW to evaluate its performance.

[1] `http://www.tpc.org/tpcds/`

REFERENCES

[1] Chandra, P.—Gupta, M. K.: Comprehensive Survey on Data Warehousing Research. International Journal of Information Technology, Vol. 10, 2018, No. 2, pp. 217–224, doi: 10.1007/s41870-017-0067-y.

[2] Santos, M. Y.—Costa, C.—Galvão, J.—Andrade, C.—Pastor, O.—Marcén, A. C.: Enhancing Big Data Warehousing for Efficient, Integrated and Advanced Analytics. In: Cappiello, C., Ruiz, M. (Eds.): Information Systems Engineering in Responsible Information Systems (CAiSE 2019). Springer, Cham, Lecture Notes in Business Information Processing, Vol. 350, 2019, pp. 215–226, doi: 10.1007/978-3-030-21297-1_19.

[3] Ng, R. T.—Han, J.: CLARANS: A Method for Clustering Objects for Spatial Data Mining. IEEE Transactions on Knowledge and Data Engineering, Vol. 14, 2002, No. 5, pp. 1003–1016, doi: 10.1109/TKDE.2002.1033770.

[4] Nebot, V.—Berlanga, R.: Building Data Warehouses with Semantic Data. Proceedings of the 2010 EDBT/ICDT Workshops (EDBT'10), 2010, doi: 10.1145/1754239.1754250.

[5] Boumlik, A.—Soussi, N.—Bahaj, M.: SMART-ETL-MR: Novel ETL Framework for Building Data Warehouse from Big Data Source Using MapReduce. Journal of Theoretical and Applied Information Technology, Vol. 98, 2020, No. 17, pp. 3449–3460.

[6] Yangui, R.—Nabli, A.—Gargouri, F.: Towards Data Warehouse Schema Design from Social Networks – Dynamic Discovery of Multidimensional Concepts. Proceedings of the 17th International Conference on Enterprise Information Systems – Volume 2 (ICEIS 2015), 2015, pp. 338–345, doi: 10.5220/0005383903380345.

[7] Dehdouh, K.: Building OLAP Cubes from Columnar NoSQL Data Warehouses. In: Bellatreche, L., Pastor, Ó., Almendros Jiménez, J. M., Aït-Ameur, Y. (Eds.): Model and Data Engineering (MEDI 2016). Springer, Cham, Lecture Notes in Computer Science, Vol. 9893, 2016, pp. 166–179, doi: 10.1007/978-3-319-45547-1_14.

[8] Llave, M. R.: Data Lakes in Business Intelligence: Reporting from the Trenches. Procedia Computer Science, Vol. 138, 2018, pp. 516–524, doi: 10.1016/j.procs.2018.10.071.

[9] Berkani, N.—Bellatreche, L.—Khouri, S.—Ordonez, C.: Value-Driven Approach for Designing Extended Data Warehouses. In: Song, I. Y., Romero, O., Wrembel, R. (Eds.): Design, Optimization, Languages and Analytical Processing of Big Data (DOLAP 2019). CEUR Workshop Proceedings, Vol. 2324, 2019.

[10] Khouri, S.—Berkani, N.—Bellatreche, L.—Lanasri, D.: Data Cube Is Dead, Long Life to Data Cube in the Age of Web Data. In: Madria, S., Fournier-Viger, P., Chaudhary, S., Reddy, P. K. (Eds.): Big Data Analytics (BDA 2019). Springer, Cham, Lecture Notes in Computer Science, Vol. 11932, 2019, pp. 44–64, doi: 10.1007/978-3-030-37188-3_4.

[11] Berkani, N.—Bellatreche, L.—Khouri, S.—Ordonez, C.: The Contribution of Linked Open Data to Augment a Traditional Data Warehouse. Journal of Intelligent Information Systems, Vol. 55, 2020, No. 3, pp. 397–421, doi: 10.1007/s10844-020-00594-w.

[12] Yangui, R.—Nabli, A.—Gargouri, F.: Automatic Transformation of Data Warehouse Schema to NoSQL Data Base: Comparative Study. Procedia Computer Science, Vol. 96, 2016, pp. 255–264, doi: 10.1016/j.procs.2016.08.138.

[13] Chevalier, M.—Malki, M. E.—Kopliku, A.—Teste, O.—Tournier, R.: Implementation of Multidimensional Databases in Column-Oriented NoSQL Systems. In: Morzy, T., Valduriez, P., Bellatreche, L. (Eds.): Advances in Databases and Information Systems (ADBIS 2015). Springer, Cham, Lecture Notes in Computer Science, Vol. 9282, 2015, pp. 79–91, doi: 10.1007/978-3-319-23135-8_6.

[14] Dehdouh, K.—Bentayeb, F.—Boussaid, O.—Kabachi, N.: Using the Column Oriented NoSQL Model for Implementing Big Data Warehouses. Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA '15), The Steering Committee of the World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2015, pp. 469–475.

[15] Prakash, D.: NOSOLAP: Moving from Data Warehouse Requirements to NoSQL Databases. Proceedings of the 14th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2019), 2019, pp. 452–458, doi: 10.5220/0007748304520458.

[16] Scabora, L. C.—Brito, J. J.—Ciferri, R. R.—de Aguiar Ciferri, C. D.: Physical Data Warehouse Design on NoSQL Databases – OLAP Query Processing over HBase. Proceedings of the 18th International Conference on Enterprise Information Systems – Volume 1 (ICEIS), SciTePress, 2016, pp. 111–118, doi: 10.5220/0005815901110118.

[17] Mior, M. J.—Salem, K.—Aboulnaga, A.—Liu, R.: NoSE: Schema Design for NoSQL Applications. IEEE Transactions on Knowledge and Data Engineering, Vol. 29, 2017, No. 10, pp. 2275–2289, doi: 10.1109/TKDE.2017.2722412.

[18] Boussahoua, M.—Boussaid, O.—Bentayeb, F.: Logical Schema for Data Warehouse on Column-Oriented NoSQL Databases. In: Benslimane, D., Damiani, E., Grosky, W. I., Hameurlain, A., Sheth, A., Wagner, R. R. (Eds.): Database and Expert Systems Applications (DEXA 2017). Springer, Cham, Lecture Notes in Computer Science, Vol. 10439, 2017, pp. 247–256, doi: 10.1007/978-3-319-64471-4_20.

[19] Boussahoua, M.—Bentayeb, F.—Boussaid, O.—Kabachi, N.: A Data Partitioning Optimization Approach for Distributed Data Warehouses on Column Family NoSQL Systems. Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA '18), The Steering Committee of the World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2018, pp. 54–60.

[20] Soni, K. G.—Patel, A.: Comparative Analysis of K-Means and K-Medoids Algorithm on IRIS Data. International Journal of Computational Intelligence Research, Vol. 13, 2017, No. 5, pp. 899–906.

[21] Boumlik, A.—Soussi, N.—Bahaj, M.: Automatic Data Modeling Transformation Approach of NoSQL Document and Column Stores to RDF. Journal of Theoretical and Applied Information Technology, Vol. 96, 2018, No. 15.

[22] Verma, R.—Puntambekar, D. M.: Comparison of Partitioning Algorithms for Categorical Data in Cluster. International Journal of Engineering Science and Com-

puting, Vol. 8, 2018, No. 7, Art. No. 18701.

[23] Strauch, C.: NoSQL Databases. Lecture Notes, Stuttgart Media University, Vol. 20, 2011, No. 24, pp. 1–149.

[24] Derrar, H.—Boussaid, O.—Ahmed-Nacer, M.: An Objective Function for Evaluation of Fragmentation Schema in Data Warehouse. Encyclopedia of Information Science and Technology, Third Edition, IGI Global, 2015, pp. 1949–1957, doi: 10.4018/978-1-4666-5888-2.ch188.

Nassima Soussi is an Assistant Professor in the Department of Mathematics and Computer Sciences from National School of Applied Sciences, Sultan Moulay Slimane University (Khouribga, Morocco). She obtained her special higher studies degree in software engineering from the National School of Applied Sciences (Khouribga, Morocco) in 2014 and Ph.D. from the Faculty of Sciences and Technology, Hassan 1st University (Settat, Morocco) in 2018. Her main research domains are semantic web and big data warehousing.