Computing and Informatics, Vol. 42, 2023, 1037–1059, doi: 10.31577/cai_2023_5_1037

AUTOSCALING METHOD FOR DOCKER SWARM TOWARDS BURSTY WORKLOAD

Qichen Huang, Song Wang, Zhijun Ding*

Department of Computer Science and Technology
Tongji University
No. 4800, Caoan Highway
Shanghai, China
e-mail: zhijun_ding@outlook.com

Abstract. The autoscaling mechanism of cloud computing can automatically adjust computing resources according to user needs, improve quality of service (QoS) and avoid over-provision. However, the traditional autoscaling methods suffer from oscillation and degradation of QoS when dealing with burstiness. Therefore, the autoscaling algorithm should consider the effect of bursty workloads. In this paper, we propose a novel AmRP (an autoscaling method that combines reactive and proactive mechanisms) that uses proactive scaling to launch some containers in advance, and then the reactive module performs vertical scaling based on existing containers to increase resources rapidly. Our method also integrates burst detection to alleviate the oscillation of the scaling algorithm and improve the QoS. Finally, we evaluated our approach with state-of-the-art baseline scaling methods under different workloads in a Docker Swarm cluster. Compared with the baseline methods, the experimental results show that AmRP has fewer SLA violations when dealing with bursty workloads, and its resource cost is also lower.

Keywords: Cloud computing, autoscaling, bursty workload, container, service-level agreement

Mathematics Subject Classification 2010: 68-W27

* Corresponding author

1 INTRODUCTION

In recent years, the rapid development of cloud computing has provided basic support for Big Data [1], Internet of Things [2], Artificial Intelligence [3], and other fields. Autoscaling is one of the important characteristics of cloud computing [4, 5], which automatically adjusts computing resources based on service requirements and preset policies. Appropriate computing resources can be allocated in a peak or a trough period of workload. Therefore, autoscaling further reflects the advantages of pay-as-you-go in cloud computing. Cloud vendors such as AWS[1], Google Cloud[2] and Microsoft Azure[3] have corresponding scaling strategies. According to the scaling policy, autoscaling can be divided into horizontal and vertical scaling [6]. Horizontal scaling refers to scaling in/out, adjusting only the number of containers/VMs (Virtual Machines). Vertical scaling refers to scaling up/down, which only adjusts the resources, such as CPU, memory, and network bandwidth. In Kubernetes [7, 8, 9], HPA [10] and VPA[4] are the horizontal and vertical scalers in the cluster, respectively. In addition, autoscaling can also be classified by scaling timing [11]. Reactive autoscaling uses the current service status and workload to make scaling decision and proactive autoscaling employs the future status of the service or workload.

Autoscaling can adjust the computing resource in real time as the workload changes. However, for bursty workload, whether it is reactive autoscaling or proactive autoscaling, there will be a period of QoS degradation. The impact of burstiness on the scaling algorithm is mainly due to two points. The first point is that the bursty workload usually fluctuates wildly, which brings oscillation to the scaling algorithm. That is, the resource provided is frequently changed; Another impact is that burstiness will cause a period of service degradation. For reactive scaling, resources are already under-provision when burstiness is detected. Similarly, proactive scaling presents a similar problem, as the quality of service (QoS) inevitably degrades when dealing with the bursty workload since it is hard to predict.

Most of the existing research about autoscaling focus on the prediction and resource provision models [12, 13, 14, 15]. For a non-bursty workload, optimizing the above two models can ensure the QoS and use fewer resources. However, autoscalers require additional optimization for bursty workloads. In this paper, we propose a novel burst-aware scaling method named AmRP (an autoscaling method that combines reactive and proactive mechanisms). AmRP can be divided into two main modules: the proactive module and the reactive module. The proactive module launches a part of the containers in advance. These containers mainly serve the reactive scaling modules. When the reactive module of AmRP performs scaling, vertical scaling can be performed on existing containers, which increases resources

[1] `https://aws.amazon.com/cn/autoscaling/`

[2] `https://cloud.google.com/compute/docs/autoscaler`

[3] `https://azure.microsoft.com/en-us/features/autoscale`

[4] `https://github.com/kubernetes/autoscaler/tree/master/vertical-pod-autoscaler`

more rapidly than horizontal scaling. In addition, burst detection is added to the reactive module. If the surge in the number of requests is detected, the scaling scheme will be further adjusted to alleviate the oscillation of the scaling algorithm and better ensure the QoS. In this paper, an AmRP prototype is designed and developed on Docker Swarm Cluster. Compared with the baseline methods, resource cost and SLA violation are lower when dealing with the bursty workload. The main contributions of this paper are as follows:

1. It designs a complete scaling system, which adjusts resources according to the real time workload to meet the expected response time.
2. It proposes a novel scaling algorithm combining reactive and proactive scaling to provide resources rapidly. The algorithm also adds burst detection to alleviate the impact of bursty workload on QoS.
3. It implements an AmRP prototype on Docker Swarm and evaluates it with the baseline methods under different workload types.

The paper is organized as follows. Section 2 introduces the work related to autoscaling in recent years. Section 3 describes in detail the AmRP scaling strategy proposed in this paper. The specific experimental design, including baseline methods and benchmark application, is described in Section 4. Section 5 analyzes the experimental results. Section 6 describes the conclusion and future work.

2 RELATED WORK

Autoscaling can be divided into reactive scaling and proactive scaling according to scaling timing. [16, 17, 18] are all about reactive scaling based on rules and analytical models. Using native Kubernetes HPA requires certain experience to set reasonable scaling rules. Therefore, [16] solves this problem employing two-stage scaling. Libra autoscaler is an autoscaler proposed by the authors. Libra first uses vertical scaling to find the optimal resource allocation for pods and then enters horizontal scaling to cope with fluctuating workloads. In [17], the author proposes a dynamic multi-layer indicator scaling method, adding application-level indicators based on native Kubernetes HPA, optimizing resource usage, and further ensuring QoS. [18] proposes a combined scaling method named COPA. When making scaling decisions, vertical and horizontal scaling are combined, and the rolling update parameter in Kubernetes is taken into account. While ensuring QoS, COPA reduces overall resource costs. The above work takes the current workload and service status as input when making scaling decisions, called reactive scaling. When proactive scaling makes scaling decisions, the input is the future workload or future service status. [19] proposes an autoscaler based on machine learning by using time series forecasting and queuing theory, which can accurately predict the workload of distributed servers, estimate resources required, optimize service response time, and meet SLA. In [20], authors use LSTM as a prediction model to dynamically scale horizontally and vertically to improve the end-to-end latency of the service.

According to the scaling policy, autoscaling can also be divided into horizontal and vertical scaling. [21] improves the VPA of Kubernetes. The problem with the native VPA is that the way to adjust resources is to start new pods and terminate old ones. Therefore, the author proposes RUBAS, which solves the VPA adjustment resource restart problem and improves resource utilization through container integration migration and checkpoint technology. The scenario in [22] is that container-based IoT applications in edge computing need to dynamically adjust resources according to the amount of IoT device requests. However, the native Kubernetes HPA evenly deploys pods on each node without considering the imbalance of resource demand among nodes in the edge computing environment. Therefore, the author proposed THPA, running on Kubernetes, to achieve real time traffic awareness and autoscaling pods for IoT applications in edge computing environments.

The impact of bursty workloads is rarely considered when scaling decisions are made. [23] uses Bi-LSTM to predict the number of HTTP requests and designs a proactive autoscaling approach in Kubernetes. Simple handling of the bursty workload is added to this method. The idea is to reserve part of pods each time when performing scaling in so that the autoscaler can have a better QoS and increase resources faster dealing with burstiness. [24] adds online burst detection into proactive autoscaling, uses standard deviation and sliding window to detect burstiness, and allocates a relatively stable amount of resources after detecting burstiness for the first time, which solves the problem of scaling oscillation, and a certain extent guaranteed QoS. The method in [25] is similar to that in [24], but the difference lies in that the information entropy method is used in [25] for burst detection. The existing scaling methods are quite simple to deal with the bursty workload. In scaling decisions, more resources are allocated to cope with burstiness that may occur at any time, resulting in higher resource costs than algorithms that do not consider bursty workload. In addition, because the bursty workload is hard to predict, the resources should be increased rapidly when burstiness is detected. However, this is rarely considered in the existing scaling strategies. Therefore, we propose a novel method combining proactive and reactive scaling. Proactive scaling performs workload prediction and launches containers in advance. Reactive scaling performs online burst detection and prioritizes vertical scaling. Compared with horizontal scaling, the vertical scaling strategy can increase resources more quickly and further reduce SLA violations.

3 PROPOSED SYSTEM

This section begins with an overview of AmRP. Section 3.2 describes the reactive module, including burst detection, resource provision model, and reactive scaling algorithm. Section 3.3 introduces the proactive module, including the time series forecasting model, an estimation of the maximum requests, and the proactive scaling algorithm.

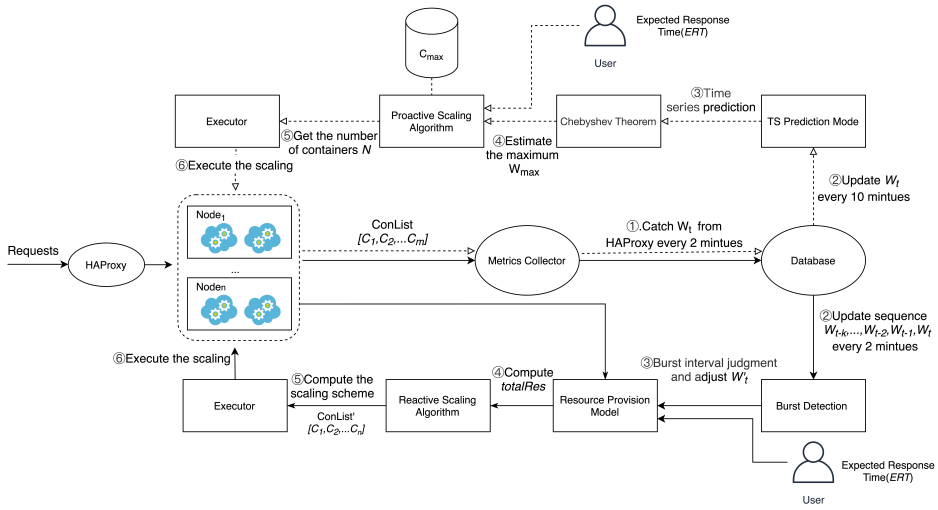


Figure 1. Overall architecture of AmRP

3.1 Overview of AmRP

The execution process of AmRP is shown in Figure 1. AmRP is an autoscaler that focuses on resource allocation at the container level. In Docker Swarm, the resources of containers are determined by the limit property. Therefore, the purpose of AmRP is to determine the number of containers and the resource setting. AmRP periodically executes the reactive module and the proactive module. The reactive module is implemented every short period, obtains the current number of requests from HAProxy in real time and performs burst detection. HAProxy[5] is a High-Performance TCP/HTTP Load Balancer. If it is in the burst interval at this time, the number of requests will be adjusted. The adjusted requests are used as input to the resource provision model. The resource provision model uses the number of requests and the expected response time from users as input to get the scaling scheme. If existing containers can be scaled vertically to meet the workload, AmRP will prioritize vertical scaling. If not, a combined scaling, that is, vertical and horizontal scaling is performed. The proactive module is executed every more extended period. It will perform time series prediction and estimation of the maximum number of requests used as one of the inputs to the proactive scaling algorithm. Only the number of containers is concerned in the proactive module.

[5] `https://www.haproxy.org`

3.2 Reactive Module

Reactive scaling module obtains the current number of requests W_t and the container list *ConList* containing resource settings for existing containers every 2 minutes and updates them to the database. The burst detection module obtains the requests sequence $W_{t-k}, \ldots, W_{t-1}, W_t$ from the database. The sequence will be judged, and the number of requests will be adjusted to better deal with the bursty workload if it is currently in the burst interval. The revised number of requests W'_t, the expected response time *ERT* set by users, and the container list *ConList* are used as the input of the resource provision model, and the output of the model is the total resource *totalRes*. The reactive module prefers vertical scaling when making scaling decisions because vertical scaling can increase resources faster than horizontal scaling.

3.2.1 Burst Detection

Workload burstiness is usually detected and judged by entropy-based methods [26, 27], but these methods are generally offline models and require a complete workload trace. For autoscaling, it is necessary to detect business in real time. In this paper, we choose an online model for burst detection. Abdullah et al. [24] also chose online bustiness detection, which detection standard was the standard deviation of sliding windows. The method adopted in this paper is the strategy of combining sliding window and boxplot [28], which can be divided into the following two steps:

1. Conduct surge point detection and judgment.
2. Determine the burst interval according to whether it is a surge point.

Lines 1 to 13 of Algorithm 1 show the function of surge point judgment. The first loop is to calculate a new time series S. It takes the mean value of the sliding window as the reference value ref_i, then takes the value l_i behind the sliding window and makes a difference between the two values. Line 8, on the new sequence S, calculates the boxplot, where Q_3 is the third percentile, IQR is the Interquartile Range, and c is the coefficient of IQR. Through the boxplot, we can get the upper bound. If it exceeds the upper bound, W_t is the surge point. Lines 15 to the end of Algorithm 1 determine the burst interval. Line 15 calls the function to determine the surge point. If the current is the surge point, then update *burstLen* to the preset length *Len*, indicating that the time points from the current point to the following length are burst intervals, and adjust the number of requests to W'_t. The reason for using this strategy to determine the burst interval is that the burstiness usually occurs continuously. Lines 20 to 27 represent that another judgment is required if it is not a surge point. When *burstLen* is greater than zero, it means that it is in the burst interval at this time, decrement the value by one and adjust the number of requests W'_t; otherwise, *burstLen* is set to zero.

Algorithm 1 Burst detection

Input: Remaining burst interval length ($burstLen$), time series ($W_{t-k}, \ldots, W_t$), window size (ws)

Output: $burstLen$

function SURGE POINT JUDGMENT($ws, W_{t-k}, \ldots, W_t$)
    $Flag \leftarrow false$
    for each i *in* $k - ws$ **do**
        $ref_i \leftarrow$ the average of the sliding window
        $l_i \leftarrow$ next value of sliding window
        $S_i \leftarrow ref_i - l_i$ // S is a new sequence
    end for
    $UpperBound \leftarrow Q_3 + c * IQR$
    if $S_t > UpperBound$ **then**
        $Flag \leftarrow true$
    end if
    return $Flag$
end function

$Flag \leftarrow$ SURGE POINT JUDGMENT($ws, W_{t-k}, \ldots, W_t$)
if $Flag = true$ **then**
    $burstLen \leftarrow Len$
    $W'_t \leftarrow max(W_{t-k}, \ldots, W_t)$
else
    if $burstLen > 0$ **then**
        $burstLen \leftarrow burstLen - 1$
        $W'_t \leftarrow max(W_{t-k}, \ldots, W_t)$
    else
        $burstLen \leftarrow 0$
        $W'_t \leftarrow W_t$
    end if
end if
return $burstLen, W'_t$

3.2.2 Resource Provision Model

Containers can be deployed without setting resource allocation. However, it will bring about resource competition between containers. Therefore, whether in Kubernetes or Docker Swarm, it is best to give the resource setting of each container. In this paper, the resource limit for a single container ranges from C_{min} to C_{max}. When a single container is scaled vertically, resources are adjusted according to fixed step size, such as adding 0.25 vCPU or 0.5 vCPU. When the resource of a single container reaches C_{max}, horizontal scaling will be executed to increase the resources.

In Resource Provision Model, the total CPU resources required will be solved. Then the specific scaling scheme will be given by the reactive scaling algorithm. The model's input is the revised number of requests W'_t, and expected response time ERT, and the model's output is the total resource amount $totalRes$.

$$\varphi : (W'_t, ERT) \rightarrow totalRes. \tag{1}$$

To collect trace data, we deploy the benchmark application (see Section 4.2) in the Docker Swarm. Then, *Hey Load Generator*[6] is used to simulate users sending requests. Then, we increase the number of requests linearly while adding resources with a reactive scaling strategy. For example, 0.25 vCPU is incrementally added to the existing container when the response time exceeds ERT. If all existing containers have reached C_{max}, the new container with C_{min} will be started. Finally, we will filter the data whose response time exceeds ERT, and the remaining data will be used for model training. Table 1 shows part of the trace data. The first row in the table indicates that when the number of requests is 804 and the total CPU resources is 10.5 vCPU, the response time is 0.1751 s.

Request	**Response Time**	**totalRes**
804	0.1751	10.5
895	0.1904	10.75
956	0.1958	11.25
...	...	...

Table 1. Trace data

In AmRP, we choose Random forest as the resource provision model. Random forest is an ensemble learning method for classification and regression that operates by constructing a multitude of decision trees at training time [29, 30]. For classification tasks, the output of the random forest is the class selected by most trees. In this paper, we use it to solve regression tasks, so the mean or average prediction of the individual trees is returned. Random decision forests correct decision trees' habit of overfitting their training set. In addition, the advantage of random forest is that its training speed is relatively fast, and it can balance errors for imbalanced data sets.

3.2.3 Reactive Scaling Algorithm

In the reactive module of AmRP, when performing scaling decisions, if vertical scaling can meet the current workload, vertical scaling is preferred. Compared to horizontal scaling, the execution of vertical scaling has a shorter duration. Therefore when burstiness is detected, vertical scaling increases resources and restores the response time to the ERT more quickly. Two methods detect the surge at Point 9 in Figure 2. Response time for vertical scaling recovers below ERT faster than for

[6] `https://github.com/rakyll/hey/releases`

horizontal scaling. Therefore, AmRP tends to perform vertical scaling in the reactive module. AmRP uses a combination of vertical scaling and horizontal scaling to increase resources only when existing containers cannot be scaled vertically. In the proactive module of AmRP, by predicting and estimating the number of containers needed in the future, containers with C_{min} resources are started in advance to ensure that the reactive module can perform vertical scaling in most cases.

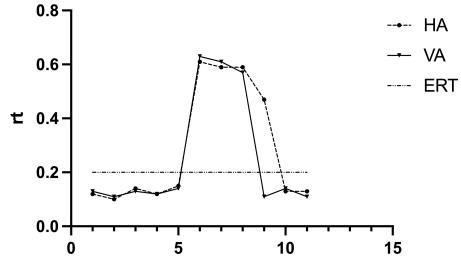


Figure 2. VA vs HA

Algorithm 2 describes the reactive scaling algorithm. The algorithm's input is the container information *ConList*, the adjusted number of requests W'_t, the expected response time *ERT*, and the output is the updated *ConList*'. Initially, the *ResourceProvisionModel* resolves the total current resource needs. Lines 2 to 24 of the algorithm are situations where it is necessary to increase the resources through vertical or combined scaling. $Count * (ConList) * C_{max}$ refers to the maximization of container resources that can reach through vertical scaling. Lines 3 to 14 indicate that adding resources to existing containers can meet the current workload. In order to adjust fewer containers when executing scaling, AmRP sorts *ConList* in ascending order according to the allocated resources and then increases the resources one by one. Lines 15 to 24 adopt a combined scaling strategy. First, set all existing containers to C_{max}. Then, the remaining required resource is added by starting new containers. In this paper, the new container's resource setting is C_{min}. Line 25 to the end of the algorithm is the case of reducing the resources. In the reactive module, only vertical scaling is used to reduce the resources. Since reserving a certain number of containers is conducive to vertical scaling, which increases resources more quickly. The operation of removing containers is performed in the proactive module. Similarly, to adjust only a tiny part of containers as much as possible, sort *ConList* in descending order and then reduce the resources one by one.

3.3 Proactive Module

The proactive module is executed every 10 minutes. The module first uses the *ARIMA* model to perform a multi-step time series prediction. Then, the predicted data is combined with a part of the historical data to form a new time series. The *Chebyshev's Inequality* is used to estimate the maximum number of requests. The

Algorithm 2 Reactive Scaling Algorithm

Input: Current container list ($ConList$), Adjusted Request at time t (W'_t), Expected Response Time (ERT)

Output: $ConList$'

```
totalRes ← ResourceProvisionModel(W'_t, ERT)
if totalRes > sum(ConList) then
    if count(ConList) * C_max <= totalRes then
        sort ConList by ascending order
        addRes ← totalRes − sum(ConList)
        for i ∈ Conlist do
            diff ← C_max − C_i
            if addRes >= diff then
                C_i ← C_max
            else
                C_i ← C_i + addRes
                break
            end if
            addRes ← addRes − diff
        end for
    else
        for i ∈ ConList do
            C_i ← C_max
        end for
        addRes ← totalRes − count(ConList) * C_max
        num ← ⌈addRes/C_min⌉
        for 1 ... num do
            ConList.append(C_min)
        end for
    end if
else if totalRes < floor * sum(ConList) then
    sort ConList by descending order
    removeRes ← sum(ConList) − totalRes
    for i ∈ ConList do
        diff ← C_i − C_min
        if removeRes >= diff then
            C_i ← C_min
        else
            C_i ← C_i − removeRes
            break
        end if
        removeRes ← removeRes − diff
    end for
end if
ConList' ← ConList
return ConList'
```

proactive scaling algorithm takes the maximum number of requests W_{max}, expected response time ERT, and the container C_{max} as inputs to obtain the number of containers required. Therefore, when the workload fluctuates wildly, AmRP will start enough containers in advance and then use vertical scaling to adjust resources in the reactive module. In addition, this module is also responsible for removing containers. When the number of requests decreases and becomes stable, AmRP performs scaling in to remove part of the containers.

3.3.1 Prediction Model

The prediction model in the proactive module is the *ARIMA* model commonly used in statistics [31, 32, 33]. *ARIMA* is widely used in time series forecasting. In $ARIMA(p, d, q)$, AR is autoregression, parameter p is the number of autoregressive terms; MA is moving average, parameter q is the number of moving average terms; I represent difference, parameter d is the number of differences to convert the non-stationary sequence into a stationary sequence. In the model, the value of the time series at the next moment is predicted based on the value observed in the past and random error. The specific formula is as follows, y_t represents the value at time t, and ε_t represents the random error at time t; φ_1 $(i = 1, 2, \ldots, p)$ and θ_j $(j = 1, 2, \ldots, q)$ are the coefficients of the AR and MA models, respectively.

$$y_t = \theta_0 + \varphi_1 y_{t-1} + \cdots + \varphi_p y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \cdots - \theta_q \varepsilon_{t-q}. \tag{2}$$

3.3.2 Chebyshev's Inequality

AmRP uses *Chebyshev's Inequality* to estimate the maximum number of requests W_{max} in the proactive module [34, 35]. This inequality generally applies to data of various distributions and is called *Chebyshev's Theorem*. The portion of any dataset that lies within k standard deviations of its mean is always at least $1 - \frac{1}{k^2}$, where k is any positive number greater than 1. When $k = 3$, it means that at least 88.9 % of all data is within three standard deviations of the mean. Therefore *Chebyshev's Inequality* can estimate the probability of an event if the distribution of the random variable X is unknown. The formula follows, where μ is the mean and σ is the standard deviation. $\mu + k * \sigma$ will be used as the estimated maximum W_{max} as one of the inputs to the proactive scaling algorithm in Section 3.3.3.

$$P(|x - \mu| \geq k\sigma) \leq \frac{1}{k^2}, \tag{3}$$

$$W_{max} \approx \mu + k * \sigma. \tag{4}$$

3.3.3 Proactive Scaling Algorithm

Algorithm 3 describes the proactive scaling algorithm. The input is container information *ConList*, the estimated maximum number of requests W_{max}, the maximum

resources of a single container C_{max}, and the expected response time ERT. The output of the algorithm is the updated $ConList'$. The algorithm begins with calculating the required total resource $totalRes$ under W_{max}. Line 2 estimates the required number of containers, which guarantees that the resources can be increased through vertical scaling in the reactive module. In lines 3 to 6 of the algorithm, AmRP will start additional containers, and the containers' resource is C_{min}. Lines 7 to 12 of the algorithm are used to remove excess containers. Similarly, $ConList$ is sorted first, then AmRP will terminate containers with relatively few resources.

Algorithm 3 Proactive Scaling Algorithm

Input: $ConList$, W_{max}, C_{max}, ERT
Output: $ConList'$

```
1: totalRes ← ResourceProvisionModel(W_max, ERT)
2: num ← ⌈totalRes/C_max⌉
3: if num > count(ConList) then
4:     for i in 1 ... num − count(ConList) do
5:         ConList.append(C_min)
6:     end for
7: else if num < count(ConList) then
8:     sort ConList by ascending order
9:     for i in 1 ... num − count(ConList) do
10:        ConList.remove(C_i)
11:    end for
12: end if
13: ConList' ← ConList
14: return ConList'
```

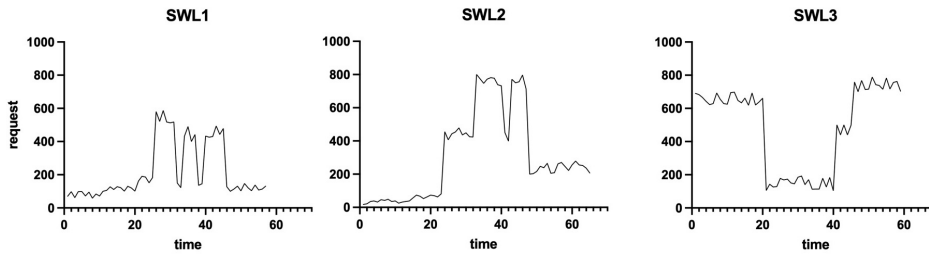
4 EXPERIMENT DESIGN

To evaluate our method, we design an AmRP prototype on the Docker Swarm platform and compare it with baseline scaling methods. All three scaling methods scale benchmark applications under various workloads. Methods are evaluated according to average response time, resource usage, and other indicators.

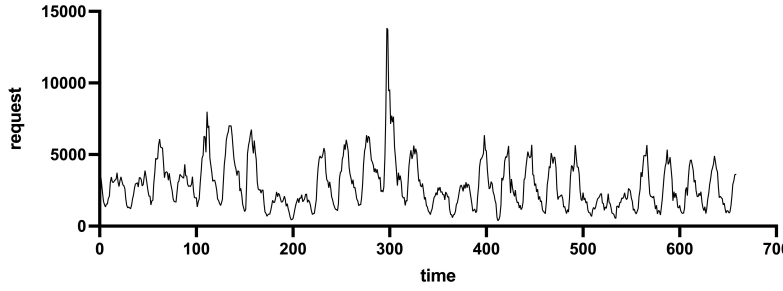
4.1 Baseline Methods

4.1.1 Base-Aware Predictive Autoscaling (BPA)

Abdullah et al. [24] proposed a burst-aware method based on proactive autoscaling, which is referred to as BPA in the rest of this paper. BPA adds burst detection based on standard deviation and sliding windows to traditional proactive autoscaling to measure burstiness in real time. If burstiness is detected, the number of instances is modified to the maximum number of instances in a nearby period. The regression



a) synthetic workloads



b) NASA workload

Figure 3. Workloads

model is used to predict the workload and the decision tree model is used to solve the scaling scheme. BPA provides a relatively stable resources when dealing with the workload with frequent fluctuations, which can better ensure the QoS and alleviate oscillation.

4.1.2 Reactive Method Based Queuing Theory (RMQ)

This baseline method is reactive autoscaling based on the Queuing Theory. In the remaining chapters of this paper, this method is referred to as RMQ. RMQ takes the processing capacity of a single service and the current number of requests as the input of the Queuing Theory model to analyze and solve and then obtains a reasonable number of instances in real time. For a non-bursty workload, this method can guarantee QoS and use fewer resources.

4.2 Benchmark Application

With the rise of microservices architecture, cloud services are becoming more and more fine-grained, such as image search, image recognition, document translation, video or audio decoding tasks, etc. Benchmark application selected in this paper

is a matrix operation, which is implemented based on PHP and packaged into containers to run in clusters. This application is a typical CPU-intensive task. In the cloud environment, most microservices are more sensitive to CPU resources, so this benchmark application is quite representative.

4.3 Simulation of Users

In the experiment, it is necessary to simulate the continuous and concurrent requests of users. In this paper, we choose to use the stress testing tool to simulate requests from users. *Hey Load Generator*[7] is an open-source stress-testing tool developed based on the Go language. To better verify the performance of the scaling algorithm under different workloads, this paper extracts some workloads showing burstinesses from the real data set, and also generates some SWL (Synthetic Workload). Figure 3 shows three SWLs and NASA workload [36]. It can be seen that burstiness occurs multiple times.

4.4 Evaluation Criteria

The selected evaluation indicators are resource usage, average request response time, SLA violations. The resource usage mainly refers to the usage of vCPU. The reason for not considering the memory is that the benchmark application is a CPU-intensive task with low memory requirements. Average request response time and SLA violations can reflect the overall QoS.

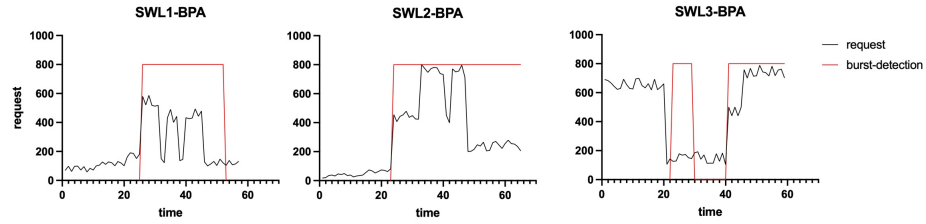
4.5 Experiment Platform

The experiment is carried out on Docker Swarm cluster, Docker version 20.10.12, where the CPU of the node is Intel(R) Xeon(R) Gold 6230 CPU @ 2.10 GHz, a total of 16 vCPU and 16 GB memory. In addition, since the stress testing tool consumes a lot of resources when sending requests concurrently, *Hey Load Generator* is independent of the experimental cluster and occupies a node with 4 vCPU and 4 GB exclusively.

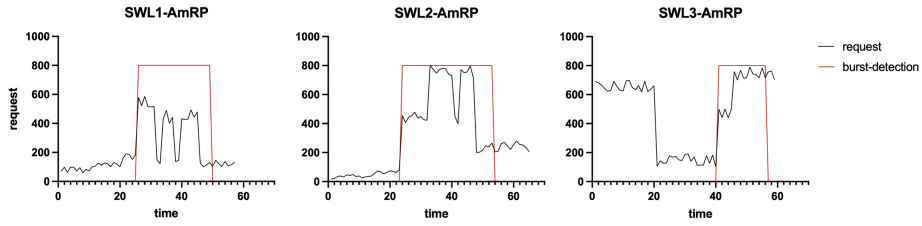
4.6 Experimental Parameters

Table 2 lists the key parameters in the experiment. There are three critical parameters in Burst Detection. First, we set k to 10, which means that we take the number of requests data adjacent to ten points as the detection input, ws is the window size of the surge point, and the setting of *Len* will affect the resource cost and QoS. If *Len* is too large, there will be some redundancy in resources, but the service quality can be better guaranteed, and vice versa. After some trial and error, we finally set this parameter to 10. Regarding container resources, we set the minimum and

[7] `https://github.com/rakyll/hey/releases`



a) Burst interval of BPA



b) Burst interval of AmRP

Figure 4. Burst detection

maximum resources of the container to 0.25 vCPU and 2 vCPU, respectively, and the step size of adjusting its resource amount is 0.25 vCPU. Since our application is a CPU-intensive service, there is very little demand for memory, and the impact of memory is not considered in this experiment. The last parameter in the table is the parameter of the *ARIMA* model.

Affiliation	**Parameters**	**Values**
Burst Detection	k	10
	ws	2
	Len	10
Container Resource	C_{min}	0.25
	C_{max}	2
	$stepSize$	0.25
ARIMA	(p, d, q)	$(2, 0, 0)$

Table 2. Part of experimental parameters

5 RESULTS

5.1 Burst Interval

AmRP and BPA both contain the burst detection module. Figure 4 shows the detection of burstiness under three SWLs by two methods. For SWL1, AmRP and BPA were almost identical in determining burst interval. AmRP set $[23, 33]$ ($Len = 10$) as the burst interval after detecting the surge point at the 23^{rd} point. Then AmRP detected two surge points, which updated the end of the burst interval twice. Therefore, for SWL1, AmRP detected the burst interval as $[23, 50]$. Similarly, BPA detected that the standard deviation of SWL1 exceeded the preset threshold at the 23^{rd} point. The standard deviation was lower than the preset threshold at the 53^{rd} point, and the number of requests at this time was lower than the moment before the start of the burst interval, so 53 was the end of the burst interval. SWL2 and SWL3 are processed differently by two methods. The determination of the burst interval by AmRP depends on the surge point and the preset length. Therefore, for SWL2, its burst interval detection is $[23, 54]$, while the detection of SWL2 by BPA belongs to burstiness from time point 23, and there is no end point. The reason is that at all time points after 23, the number of requests is higher than the moment before the first burstiness, so it is impossible to exit the burst interval. For SWL3, there is a sudden drop in the number of requests around time point 20, and since BPA uses the standard deviation, it is determined as a burstiness. The starting point of another burst interval detected by BPA is 40. Similar to SWL2, this interval also cannot be exited. The result of AmRP for SWL3 detection is that $[40, 57]$ is the burst interval. It can be seen that the burst detection of AmRP is guided by the surge point. When a surge point is detected, it enters the burst interval, and if there is no burstiness point for a long time, it exits the burst interval. The burst detection of BPA is oriented by the standard deviation, and its algorithm also marks the number of requests before the surge to exit the burst interval. For some workloads that have been stable for a long time after the burstiness, it will lead to certain waste of resources.

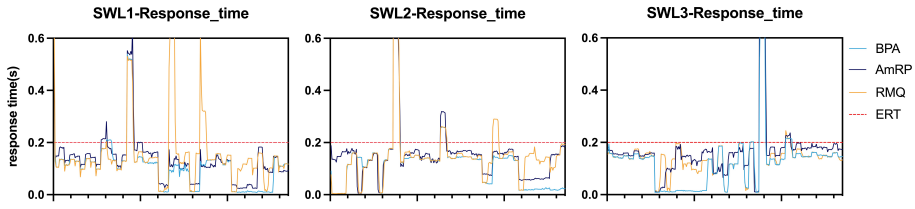


Figure 5. RT comparison under SWL

5.1.1 Response Time, SLA Violation

Figure 5. shows the response time comparison between AmRP and baseline scaling methods under SWL. To display the content of the figure more clearly, the upper limit of the response time is 0.6 s, which means that the 0.6 s in the figure may be greater than 0.6 s. There were three surges in SWL1. All scaling methods experienced a period of service degradation because the initial burstiness was difficult to predict and deal with. While AmRP and BPA benefit from burst detection, they will provide relatively stable resources once burstiness is detected. Therefore, the response time of AmRP and BPA did not exceed *ERT* in the subsequent two surges. However, the RMQ method does not consider bursty workload, so resources are frequently added/removed during burstiness, which makes its response time fluctuation obvious, and its quality of service is the worst among the three scaling methods.

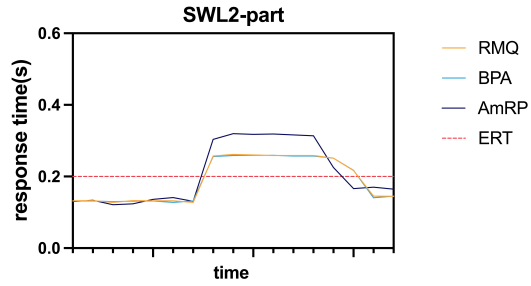


Figure 6. SWL2-Part

Similarly, in SWL2, all methods experience a drop in QoS for a while when the first surge occurs. After AmRP and BPA detected the burstiness, they were marked as surge status. A subsequent surge occurred shortly after that, and the number of requests exceeded the previous surge. At this time, it can be seen that the QoS of the three scaling algorithms will still decline because the strategy of AmRP and BPA in burst interval is to take the maximum resource amount near the time. Figure 6 shows the details of the second surge. AmRP restores the response time to *ERT* fastest among all methods. This benefits from the proactive module of AmRP that starts the container in advance. When the reactive module detects a surge, it increases resources through vertical scaling on existing containers. By comparison, the baseline method uses horizontal scaling to increase resources, so it takes more time to reduce the response time to the desired value. The three scaling ways in SWL3 and NASA (see Figure 7) are similar to SWL1 and SWL2. Table 3 summarizes the SLA violation of different methods. Except for SWL1, AmRP has the lowest SLA violation rate.

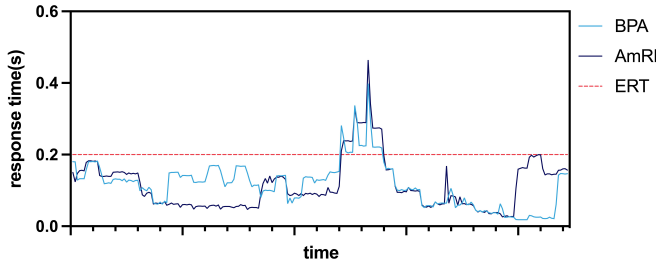


Figure 7. RT comparison under NASA

5.1.2 Resource Usage

Table 4 shows the total resource usage under various workloads. RMQ does not take bursty workload into account, and its resource usage is directly related to request trends. So RMQ is generally the scaling method that uses the least resources. AmRP and BPA give the benchmark application relatively stable resources in the burst interval, so these two methods use more resources than RMQ. The total resource usage of AmRP is lower than that of BPA, which benefits from the heterogeneity of the scaling scheme in AmRP and the finer granularity of resources. In addition, burst detection of AmRP can determine the burst interval more accurately than BPA.

Method / WL	AmRP	BPA	RMQ
SWL1	92.3 %	95.3 %	90.5 %
SWL2	95.3 %	95.1 %	92.9 %
SWL3	95.1 %	93.1 %	92.7 %
NASA	91.5 %	91 %	–

Table 3. SLA violations

Method / WL	AmRP	BPA	RMQ
SWL1	1 495	1 584	1 227
SWL2	2 227	2 400	1 830
SWL3	2 182	2 934	1 986
NASA	1 882	1 986	–

Table 4. Total resource usage

6 CONCLUSION AND FUTURE WORK

We propose a novel scaling method named AmRP for bursty workloads. AmRP is mainly divided into two parts, a proactive scaling module and a reactive scaling module. In proactive scaling, multi-step time series prediction and the maximum number of requests estimation are performed, and horizontal scaling is performed. This module aims to start enough containers in advance so that in the reactive module, vertical scaling can be executed on existing containers. The reactive module includes real time burst detection, and when calculating scaling solutions, vertical scaling is preferred. Experimental results show, compared with the baseline scaling algorithm, that for bursty workload, the AmRP scaling method further alleviates the QoS degradation caused by the surge of requests compared with BPA. As a result, AmRP can increase resources more rapidly and simultaneously provide stable resources within the burst interval. In terms of resource usage, since AmRP is a heterogeneous scaling solution, more fine-grained resource allocation further reduces the cost of AmRP resources.

Experimental results also show that the AmRP and baseline scaling methods are ineffective in dealing with the first surge or continuous surge because AmRP and BPA essentially do not predict the burstiness but only deal with them accordingly after detecting the surge. When the scaling method detects burstiness, the QoS declines, and the average response time exceeds the ERT. The solution in this paper is how to increase the resources more quickly after detecting a surge. Therefore, in future work, we will consider introducing burstiness prediction which can further improve the QoS of the scaling method in dealing with bursty workloads. However, its effect depends more on the accuracy of the burstiness prediction model.

REFERENCES

[1] Das, M.—Dash, R.: Role of Cloud Computing for Big Data: A Review. In: Mishra, D., Buyya, R., Mohapatra, P., Patnaik, S. (Eds.): Intelligent and Cloud Computing (ICICC 2019, Volume 2). Springer, Singapore, Smart Innovation, Systems and Technologies (SIST), Vol. 153, 2021, pp. 171–179, doi: 10.1007/978-981-15-6202-0_18.

[2] Sadeeq, M. M.—Abdulkareem, N. M.—Zeebaree, S. R. M.—Ahmed, D. M.—Sami, A. S.—Zebari, R. R.: IoT and Cloud Computing Issues, Challenges and Opportunities: A Review. Qubahan Academic Journal, Vol. 1, 2021, No. 2, pp. 1–7, doi: 10.48161/qaj.v1n2a36.

[3] Saadia, D.: Integration of Cloud Computing, Big Data, Artificial Intelligence, and Internet of Things: Review and Open Research Issues. International Journal of Web-Based Learning and Teaching Technologies (IJWLTT), Vol. 16, 2021, No. 1, pp. 10–17, doi: 10.4018/IJWLTT.2021010102.

[4] Coutinho, E. F.—de Carvalho Sousa, F. R.—Rego, P. A. L.—Gomes, D. G.—de Souza, J. N.: Elasticity in Cloud Computing: A Survey.

Annals of Telecommunications – Annales des Télécommunications, Vol. 70, 2015, No. 7, pp. 289–309, doi: 10.1007/s12243-014-0450-7.

[5] Chen, T.—Bahsoon, R.: Toward a Smarter Cloud: Self-Aware Autoscaling of Cloud Configurations and Resources. Computer, Vol. 48, 2015, No. 9, pp. 93–96, doi: 10.1109/MC.2015.278.

[6] Singh, P.—Gupta, P.—Jyoti, K.—Nayyar, A.: Research on Auto-Scaling of Web Applications in Cloud: Survey, Trends and Future Directions. Scalable Computing: Practice and Experience, Vol. 20, 2019, No. 2, pp. 399–432, doi: 10.12694/scpe.v20i2.1537.

[7] Burns, B.—Grant, B.—Oppenheimer, D.—Brewer, E.—Wilkes, J.: Borg, Omega, and Kubernetes. Communications of the ACM, Vol. 59, 2016, No. 5, pp. 50–57, doi: 10.1145/2890784.

[8] Casalicchio, E.: Container Orchestration: A Survey. In: Puliafito, A., Trivedi, K. S. (Eds.): Systems Modeling: Methodologies and Tools. Springer, Cham, EAI/Springer Innovations in Communication and Computing, 2019, pp. 221–235, doi: 10.1007/978-3-319-92378-9_14.

[9] Brewer, E. A.: Kubernetes and the Path to Cloud Native. Proceedings of the Sixth ACM Symposium on Cloud Computing (SoCC '15), 2015, pp. 167–167, doi: 10.1145/2806777.2809955.

[10] Nguyen, T. T.—Yeom, Y. J.—Kim, T.—Park, D. H.—Kim, S.: Horizontal Pod Autoscaling in Kubernetes for Elastic Container Orchestration. Sensors, Vol. 20, 2020, No. 16, Art. No. 4621, doi: 10.3390/s20164621.

[11] Qu, C.—Calheiros, R. N.—Buyya, R.: Auto-Scaling Web Applications in Clouds: A Taxonomy and Survey. ACM Computing Surveys (CSUR), Vol. 51, 2018, No. 4, Art. No. 73, doi: 10.1145/3148149.

[12] Etemadi, M.—Ghobaei-Arani, M.—Shahidinejad, A.: A Cost-Efficient Auto-Scaling Mechanism for IoT Applications in Fog Computing Environment: A Deep Learning-Based Approach. Cluster Computing, Vol. 24, 2021, No. 4, pp. 3277–3292, doi: 10.1007/s10586-021-03307-2.

[13] Schuler, L.—Jamil, S.—Kühl, N.: AI-Based Resource Allocation: Reinforcement Learning for Adaptive Auto-Scaling in Serverless Environments. 2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid), 2021, pp. 804–811, doi: 10.1109/CCGrid51090.2021.00098.

[14] Golshani, E.—Ashtiani, M.: Proactive Auto-Scaling for Cloud Environments Using Temporal Convolutional Neural Networks. Journal of Parallel and Distributed Computing, Vol. 154, 2021, pp. 119–141, doi: 10.1016/j.jpdc.2021.04.006.

[15] Toka, L.—Dobreff, G.—Fodor, B.—Sonkoly, B.: Machine Learning-Based Scaling Management for Kubernetes Edge Clusters. IEEE Transactions on Network and Service Management, Vol. 18, 2021, No. 1, pp. 958–972, doi: 10.1109/TNSM.2021.3052837.

[16] Balla, D.—Simon, C.—Maliosz, M.: Adaptive Scaling of Kubernetes Pods. NOMS 2020 – 2020 IEEE/IFIP Network Operations and Management Symposium, 2020, pp. 1–5, doi: 10.1109/NOMS47738.2020.9110428.

[17] Taherizadeh, S.—Stankovski, V.: Dynamic Multi-Level Auto-Scaling Rules for

Containerized Applications. The Computer Journal, Vol. 62, 2019, No. 2, pp. 174–197, doi: 10.1093/comjnl/bxy043.

[18] Ding, Z.—Huang, Q.: COPA: A Combined Autoscaling Method for Kubernetes. 2021 IEEE International Conference on Web Services (ICWS), 2021, pp. 416–425, doi: 10.1109/ICWS53863.2021.00061.

[19] Moreno-Vozmediano, R.—Montero, R. S.—Huedo, E.—Llorente, I. M.: Efficient Resource Provisioning for Elastic Cloud Services Based on Machine Learning Techniques. Journal of Cloud Computing, Vol. 8, 2019, No. 1, Art. No. 5, doi: 10.1186/s13677-019-0128-9.

[20] Marie-Magdelaine, N.—Ahmed, T.: Proactive Autoscaling for Cloud-Native Applications Using Machine Learning. GLOBECOM 2020 – 2020 IEEE Global Communications Conference, 2020, pp. 1–7, doi: 10.1109/GLOBECOM42002.2020.9322147.

[21] Rattihalli, G.—Govindaraju, M.—Lu, H.—Tiwari, D.: Exploring Potential for Non-Disruptive Vertical Auto Scaling and Resource Estimation in Kubernetes. 2019 IEEE 12th International Conference on Cloud Computing (CLOUD), 2019, pp. 33–40, doi: 10.1109/CLOUD.2019.00018.

[22] Phuc, L. H.—Phan, L. A.—Kim, T.: Traffic-Aware Horizontal Pod Autoscaler in Kubernetes-Based Edge Computing Infrastructure. IEEE Access, Vol. 10, 2022, pp. 18966–18977, doi: 10.1109/ACCESS.2022.3150867.

[23] Dang-Quang, N. M.—Yoo, M.: Deep Learning-Based Autoscaling Using Bidirectional Long Short-Term Memory for Kubernetes. Applied Sciences, Vol. 11, 2021, No. 9, Art. No. 3835, doi: 10.3390/app11093835.

[24] Abdullah, M.—Iqbal, W.—Berral, J. L.—Polo, J.—Carrera, D.: Burst-Aware Predictive Autoscaling for Containerized Microservices. IEEE Transactions on Services Computing, Vol. 15, 2022, No. 3, pp. 1448–1460, doi: 10.1109/TSC.2020.2995937.

[25] Tahir, F.—Abdullah, M.—Bukhari, F.—Almustafa, K. M.—Iqbal, W.: Online Workload Burst Detection for Efficient Predictive Autoscaling of Applications. IEEE Access, Vol. 8, 2020, pp. 73730–73745, doi: 10.1109/ACCESS.2020.2988207.

[26] Lassnig, M.—Fahringer, T.—Garonne, V.—Molfetas, A.—Branco, M.: Identification, Modelling and Prediction of Non-Periodic Bursts in Workloads. 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, 2010, pp. 485–494, doi: 10.1109/ACCESS.2020.2988207.

[27] Ali-Eldin, A.—Seleznjev, O.—Sjöstedt-de Luna, S.—Tordsson, J.—Elmroth, E.: Measuring Cloud Workload Burstiness. 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing, 2014, pp. 566–572, doi: 10.1109/UCC.2014.87.

[28] Frigge, M.—Hoaglin, D. C.—Iglewicz, B.: Some Implementations of the Boxplot. The American Statistician, Vol. 43, 1989, No. 1, pp. 50–54, doi: 10.1080/00031305.1989.10475612.

[29] Biau, G.—Scornet, E.: A Random Forest Guided Tour. Test, Vol. 25, 2016, No. 2, pp. 197–227, doi: 10.1007/s11749-016-0481-7.

[30] Liu, Y.—Wang, Y.—Zhang, J.: New Machine Learning Algorithm: Random Forest. In: Liu, B., Ma, M., Chang, J. (Eds.): Information Computing and Applications.

Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 7473, 2012, pp. 246–252, doi: 10.1007/978-3-642-34062-8_32.

[31] Liu, C.—Hoi, S. C. H.—Zhao, P.—Sun, J.: Online ARIMA Algorithms for Time Series Prediction. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 30, 2016, No. 1, pp. 1867–1873, doi: 10.1609/aaai.v30i1.10257.

[32] Ho, S. L.—Xie, M.—Goh, T. N.: A Comparative Study of Neural Network and Box-Jenkins ARIMA Modeling in Time Series Prediction. Computers and Industrial Engineering, Vol. 42, 2002, No. 2-4, pp. 371–375, doi: 10.1016/S0360-8352(02)00036-0.

[33] Mahalakshmi, G.—Sridevi, S.—Rajaram, S.: A Survey on Forecasting of Time Series Data. 2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE '16), 2016, pp. 1–8, doi: 10.1109/ICCTIDE.2016.7725358.

[34] Saw, J. G.—Yang, M. C. K.—Mo, T. C.: Chebyshev Inequality with Estimated Mean and Variance. The American Statistician, Vol. 38, 1984, No. 2, pp. 130–132, doi: 10.1080/00031305.1984.10483182.

[35] Amidan, B. G.—Ferryman, T. A.—Cooley, S. K.: Data Outlier Detection Using the Chebyshev Theorem. 2005 IEEE Aerospace Conference, 2005, pp. 3814–3819, doi: 10.1109/AERO.2005.1559688.

[36] Arlitt, M. F.—Williamson, C. L.: Internet Web Servers: Workload Characterization and Performance Implications. IEEE/ACM Transactions on Networking, Vol. 5, 1997, No. 5, pp. 631–645, doi: 10.1109/90.649565.

Qichen Huang received his B.Sc. degree in computing science and technology from Shanghai Ocean University, Shanghai, China, in 2019. He is currently pursuing the M.S. degree in the Department of Computer Science and Technology, Tongji University, Shanghai, China. His current research interests include services computing.

Song Wang received his M.Sc. degree from the Shandong University of Science and Technology, Qingdao, China, in 2016. He is currently pursuing his Ph.D. degree with the Department of Computer Science and Technology, Tongji University, Shanghai, China. His research interest is services computing.

Zhijun Ding received his M.Sc. degree from the Shandong University of Science and Technology, Tai'an, China, in 2001, and the Ph.D. degree from Tongji University, Shanghai, China, in 2007. He serves currently as Professor with the Department of Computer Science and Technology, Tongji University. He has published over 100 papers in domestic and international academic journals and conference proceedings. His research interests are in formal engineering, Petri nets, services computing, and mobile internet.

Computing and Informatics, Vol. 42, 2023, 1060–1090, doi: 10.31577/cai_2023_5_1060

CYBERSECURITY: BOTNET THREAT DETECTION ACROSS THE SEVEN-LAYER ISO-OSI MODEL USING MACHINE LEARNING TECHNIQUES

Rami Mohammed Baazeem

MIS Department, University of Jeddah
Saudi Arabia
e-mail: rbaazeem@uj.edu.sa, rami_mohammed78@outlook.com

Abstract. The Open System Interconnection (OSI) model, consisting of seven layers, has become increasingly important in addressing cyber security issues. The rapid growth of network technology has led to a rise in cyber threats, with botnets taking over fixed and mobile computers. The widespread availability of mobile devices has led to increased app consumption, with 60 % of Android malware containing major or minor botnets. The ease of accessibility of mobile devices has accelerated the adoption of mobile apps in various use cases. This article aims to identify and reduce botnets in operating systems, focusing on identifying them faster and reducing attack impact. The study analyzes botnet characteristics under controlled conditions and creates four traffic flow components across multiple time ranges. Using machine learning, flow vectors are created to identify internet flows as botnet flows or predicted flows. The method uses a combination of Boosted decision tree ensemble classifier, Naive Bayesian statistical classifier, and SVM discriminative classifier to accurately identify both well-known and novel botnets, reducing false positives and improving detection accuracy.

Keywords: Spoofing, hijacking, SYN flood, DoS, DDoS, cybersecurity threats, ISO-OSI model, machine learning

1 INTRODUCTION

The contemporary communication and computer infrastructures are very susceptible to many of various kinds of attacks. Modern communications and computing infrastructures are vulnerable to cyber-attacks like phishing, malware, DDoS, and

advanced persistent threats due to system vulnerabilities. Cloud and mobile device use has increased these risks. Robust security measures and staying up to date with security trends are crucial for organizations to mitigate these risks. The traditional starting point for these kinds of attacks is the installation of malicious software, often known as malware, which may take the form of computer viruses, worms, and Trojan horses [1]. These attacks harm on the Internet and result in many issues for users, including data corruption, delays, and severe capacity waste on the network. In addition, some of these assaults are used to obtain control of Internet hosts, which may then be used to carry out denial-of-service attacks on other websites. These attacks can be carried out in many different ways. If an attacker is successful in getting access to network hosts [2], this might result in considerable harm to the network. This damage could include the interruption of e-commerce websites, news sources, network infrastructure, routers, and root name servers, amongst other things. An adversary might potentially speed up the process of gaining access to systems and improve their chances of being successful by making use of malicious software. The term "bot" is used to refer to every one of these compromised machines. A collection of computers that have been compromised and are now referred to as bots might be referred to as a botnet [3]. Using a Command and Control (C & C) channel that has been specially set up, an intruder known as a botmaster manages these bots remotely from a faraway location. This newly discovered facet of the danger presented by botnets puts the lives of millions of people and the essential network infrastructure of countries all over the world at hazard [4]. The exponential growth of botnets poses a substantial threat to the authenticity of both computer networks and data. Botnets are a major threat to computer security, with some of the most common threats including Distributed Denial of Service (DDoS) attacks, spamming, click fraud, theft of sensitive information, spreading malware and viruses, and unauthorized access to infected machines. Consequently it is of the utmost necessity to move fast in the direction of developing a dependable detection system against botnets. This chapter begins with a brief overview of botnets, followed by a discussion of the motivation for doing the research, followed by a rundown of the various research projects that were carried out to accomplish the goals, and finally, the chapter concludes.

Open System Interconnection, commonly known as a reference model, is what is meant when someone uses the phrase "OSI model". In the year 1984, the International Organization for Standardization (ISO) was the organization responsible for developing it. This seven-layered architectural model was created to assist vendors and developers in the process of creating interoperable network systems [5].

The Open Systems Interconnection (OSI) model has seven layers: the physical layer, the media access control (MAC) layer, the network layer, the transport layer, the session layer, the presentation layer and the application layer. It offers an overview of how packets flow through the network when communication is taking place [1] and discusses the many roles that are performed by each tier. At each of these tiers, different protocols are put into operation. Using this OSI model with

seven layers, each job in the network was broken down into its component elements, which were referred to as layers. Each layer is responsible for carrying out the particular duty that has been delegated to it. Every layer in the network will provide a service to the layer above it and will make use of the services offered by the layer below it [2]. One can have a better understanding of the visual depiction of what is taking place in internet communication and how the devices are functioning inside the network with the assistance of this model.

At this point, the development of internet technology is increasing at an alarmingly quick rate, which in turn leads to high-profile hacking tactics. This OSI model not only helps one understand how network communication works, but it also enables one to examine how each layer is impacted by potential risks to one's privacy and security. A hostile act that aims to harm, steal, or disrupt electronic data and equipment are referred to as a cyber threat. The proliferation of online dangers have led to an increase in both the importance of network security and its status as a primary source of anxiety [6]. Social media can increase anxiety and depression due to factors such as social comparison, cyberbullying, fear of missing out, information overload, exposure to negative news, and addiction. It is crucial to acknowledge these triggers and take measures to minimize their adverse effects on mental health. The design of the internet itself makes it easier for many security breaches to take place. Discovering and addressing network security issues will be made easier by gaining an understanding of the vulnerabilities and different forms of attacks that might be launched against the system.

The authors of [7] researched the potential online dangers posed by social networking services. The fact that the number of people using social networking sites is growing daily provides countless opportunities for cybercriminals, who may target victims and steal critical information by using these platforms. When accessing social media networks, a person is exposed to a fast increasing number of risks to both their privacy and their security. Since Wireless Sensor Network (WSN) uses the OSI model, several kinds of attacks and countermeasures are described in [8]. These are carried out to maintain significant aspects such as confidentiality, authenticity, and integrity.

A breach of security that takes advantage of a weakness in the system or program is what we refer to as a threat. In this context, the term "vulnerability" refers to flaws or openings in the design. Vulnerability in a network may often be traced to faulty design of the network's hardware or software [9]. A danger may be either purposeful or inadvertent, such as those posed by nature or those that humans have created. An assault is always prompted in response to a threat. Unlawful activity carried out on a system is referred to as an attack. When anything causes harm, it is usually done on purpose and with malice. Controlling the vulnerabilities allows for protection against both potential threats and actual assaults. In general, attacks may be broken down into two categories:

1. the assault from inside, and
2. the attack from outside [10].

A passive attack often involves listening to and analysing the traffic on a network without interfering with the devices' ability to communicate with one another. Defending against this kind of assault may be challenging. On the other hand, an active attack is one that intentionally seeks to alter or remove the data that is being sent across the network. This attacker will often cause problems for the functioning of the network.

DoS and DDoS are the most destructive attacks that may be performed at any of the OSI model levels [11]. DoS stands for denial of service, while DDoS stands for distributed denial of service. A single system launches an assault on the victim of a DoS, however in a Distributed Denial of Service (DDoS), several bots referred to as Zombies are utilized to deliver an overwhelming volume of traffic to the victim. DDoS refers to an assault that uses a large number of dispersed devices to perform a DoS.

Security breaches and assaults on the network layer are very perilous and might bring the system's performance to a lower level overall. Because the network layer is responsible for handling source and destination IP addresses, there is a high probability that spoofing attacks will be initiated [12]. In addition, the protocols that make up the transport layer, such as TCP and UDP, contain security flaws that may be exploited to launch flooding assaults.

The "Man in the Middle" attack enables the perpetrator to listen in on and steal information from a conversation taking place between two devices on a network [13]. MITM may either be passive or active, depending on the situation. This attack may be carried out in a variety of ways, including session hijacking, ARP poisoning, and DNS spoofing, among others.

The main contribution is discussed as follows: This article aims to improve the identification and mitigation of botnet attacks on popular operating systems like Microsoft Windows and Android. By analyzing botnet characteristics under controlled conditions and using machine learning techniques, the article develops a method that accurately classifies internet flows as botnet or normal flows. The approach reduces false positives, enhances detection accuracy, and minimizes the impact of botnet attacks.

2 RELATED WORKS

Bots have the potential to be highly helpful pieces of software when they are designed to assist a human user in some manner, such as by automating an easy process or by simplifying the user's control over a range of programs and systems. During the time in question, the Eggdrop bot was widely acknowledged as a valid IRC bot and a significant number of individuals made substantial use of it in this capacity. Unfortunately, subsequent IRC bots are also built to undertake destructive activities, which compromise the system and enable the attacker to acquire sensitive information and participate in other malicious operations. This allows the attacker to forward their malicious agenda. Both the Trojan Sub7 and the worm

were responsible. Pretty Park is contending for the title of the piece of software that is credited with being the one that began the process of creating the botnet. The malware known as Sub7 is a Trojan horse, while Pretty Park is a worm. These malicious programs were the first to propose the concept of connecting to an IRC channel to listen for commands to execute harmful code [14].

This bot was developed on top of the mIRC client, which can access raw TCP and UDP sockets and run user-defined scripts in response to activities that take place in an IRC channel. Because of this, it is well suited for implementation in simple denial-of-service (DoS) 4 attacks [15]. There are at least a hundred distinct types of GTbot that are currently active and comprise an IRC client. These bots are all a member of the botnet. Before that, in 2002, the commercialization of SDBot made it feasible to build a new botnet that was built on top of existing ones. This was previously not conceivable. This bot was made up of a solitary binary file that was very short and was written in C++. The features of its backdoor make it possible for additional instructions and functions to be carried out on a machine that has been hacked. Among these operations include verifying the current condition of the infection, removing the bot from IRC [16], coming up with a random username, sending ping attacks, forcing a bot to join a channel, carrying out SYN flood or DDoS attacks, and a great deal more. In the same year, Agobot revealed the concept of a modular staged attack. It was packaged as extra malicious payloads that were put on top of an underlying backdoor. The initial phase of the attack is to install the backdoor, which prepares the way for following attacks that are segmented into several smaller modules. Once this step is complete, the route is clear for the subsequent assaults. The makeover that SDBot went through in 2003 ultimately led to it being rechristened as Spybot. These additional capabilities included the recording of keystrokes and the sending of unwanted instant messages in bulk (SPIM) [17]. Rbot, a programe that worked via the use of a SOCKS proxy, was presented to the public the same year. In addition to that, it was able to carry out distributed denial-of-service assaults and steal data using a variety of different techniques. This family of bots was the first to make use of technologies such as compression and encryption to avoid being identified.

Later on, botmasters came to the realization that IRC-based C & C servers were a single point of failure because it was very simple to delete them or just ban them, which would effectively terminate the botnet. This realization was reached because that it was very simple to do either of these things. It is possible that the IRC port has been blocked on the majority of current firewalls, and it is not difficult to identify the protocol when one examines network traffic. The botmasters were forced to alter the way how they controlled the botnets to prevent themselves from being found. They took advantage of a Peer-to-Peer, or P2P, architecture in order to construct the next generation of botnets. Within the context of P2P, a botmaster will issue an instruction to one or more bots, and those bots will then communicate the instruction to the users who are located in proximity to them. If a particular bot cannot be reached, the communication will continue with the next bot on the list. As a consequence of this, the botnet grew into a tremendously strong network,

making its elimination extremely difficult [18]. The term "hypertext transfer protocol" (HTTP) traffic refers to the kind of traffic that makes up most of all traffic on the Internet. Since that HTTP traffic is still quite common across all networks, the command-and-control servers for these botnets are still operational [19].

3 THREAT ANALYSIS

3.1 Dangers Posed by the Application Layer

The application layer in HTTPS interacts directly with the user because it manages requests and responses such as web pages, emails, and file transfers. It provides the user interface and indicates data in an approach that is understandable by individuals. HTTPS encrypts data transmission between the client and server at the application layer, enabling end-to-end encryption and secure communication between the user and server. The only layer that communicates directly with the end user is the application layer, which is also known as the presentation layer. Because it is situated in the OSI model so near to the layer that serves the end user, this layer is the most susceptible to attack. Numerous services, including e-mail, file transfers, chatting, and web surfing, are carried out at this location. The most frequent types of attacks that may take place at this layer are known as HTTP floods and SQL injections. Hackers use HTTP floods and SQL injections to gain unauthorized access to networks or systems by overloading servers or exploiting vulnerabilities in web applications. Protecting against these attacks involves implementing security measures such as firewalls, intrusion detection and prevention systems, regular security audits, and keeping software up to date. Educating users on secure computing practices is also important [20]. The objective of the Distributed Denial of Service (DDoS) attack known as an HTTP flood is to freeze the application and prevent the end user from gaining access to it. HTTP flood is a DDoS attack that overloads an application or server with excessive traffic, causing damage and financial loss. Attackers use botnets or compromised devices to launch the attack easily. Load balancing, rate limiting, and traffic filtering are mitigation measures, while regular security assessments can identify vulnerabilities. When a client wishes to connect with the server, it will often submit an HTTP request, such as GET or POST, in order to get access to the resources. This allows the client to communicate with the server.

HTTP GET Flood – A GET request is used to get and see the usual material like photos and files. This request may be used to flood websites. HTTP GET requests are important as they allow users to retrieve data or resources from a web server efficiently. GET requests are used to retrieve web pages, access APIs, and can be cached by web browsers for faster access. They are also used by search engines to index web pages, impacting a website's SEO. As a result, HTTP GET requests are a vital component of the HTTP protocol. Unauthorized sources will submit repeated HTTP GET requests to a particular server in an attempt to access the files and pictures stored on that server [21]. The server is unable to react to the

request coming from the authentic source because it is too busy dealing with the many requests that are coming from the fraudulent source.

HTTP POST – A request known as a POST is used whenever there is a need to dynamically update the resources. To bring the server down, the maximum number of resources that may be allotted to a single request will be used [22].

SQL Injection is a kind of attack that is also known as SQLI. During this assault, the attacker takes advantage of application weaknesses in order to inject malicious SQL code. This gives the attacker access to the database, enables them to edit the information that is already there, allows them to carry out administrative tasks, and even grants them access to sensitive material that is stored in the database [23].

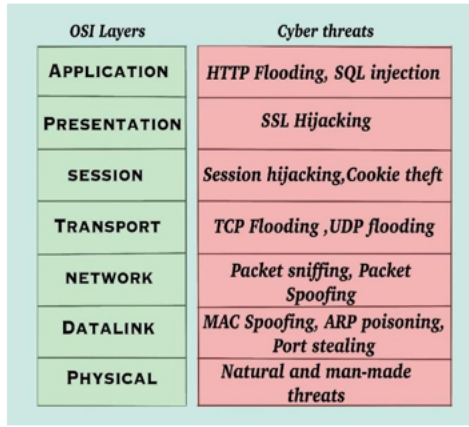


Figure 1. Cyber threats in a seven-layered OSI model

3.2 Presentation Layer Threats

This layer is also sometimes referred to as the Syntax or Translation layer. Its purpose is to provide the application layer with data that is presented in a way that is easier to understand. Encoding and decoding, data encryption and decryption, and data compression are the operations that the presentation layer is responsible for carrying out [24]. The Secure Socket Layer (SSL) is a means for encrypting data that is sent over the internet. SSL is a protocol which provides secure and encrypted connections between web servers and client browsers to avoid attackers from stealing or copying sensitive data. Data is encrypted using public key encryption, and the web server's identity is examined. SSL improves website SEO, encourages customer loyalty, and establishes trust between websites and users, all of which increase sales and revenue for businesses. SSL is prioritized over HTTP in HTTPS. SSL is not preferred over HTTPS because, due to known vulnerabilities, SSL has been deprecated in favour of TLS. HTTPS is the recommended method

for secure data transmission over the internet because it provides authentication, encryption, and trust between a web server and a user's browser. HTTPS validates the identity of a website, encrypts data transmission, and is recognized by many as a reliable and secure protocol. At the beginning of a connection between a client and a server, the server will give the client a certificate that includes the digital signature to demonstrate that it is an authorized identity [25]. Therefore, in a hijacking, the communication that is supposed to be going between the client and the server is instead intercepted by the attacker, who then attempts to pose as a genuine service. Therefore, the client and the server no longer have a secure connection, which is indicated by the HTTPS protocol [26].

3.3 Dangers at the Session Layer

Synchronization is handled at the OSI session layer, which is where the responsibility lies. This layer is responsible for the initiation of sessions between end-user application processes, as well as their management, acceptance, and closure. It also serves the purpose of a token manager by prohibiting two different users from carrying out the same activity at the same time [27].

Session hijacking is one of the risks that are present at this layer. Session hijacking occurs frequently at the OSI session layer because it manages session-related tasks, such as authentication and authorization, and uses session IDs or tokens that can be intercepted by attackers. To prevent session hijacking, it is important to use secure protocols, implement secure coding practices, and take precautions such as using strong passwords and avoiding public Wi-Fi networks. An adversary has taken control of the session and hijacked the user's session in this instance. The user has lost control of the session. Cross-site scripting is an attack technique that hackers use to take over users' sessions. An example of a client-side injection attack is known as cross-site scripting (XSS). This kind of attack involves injecting malicious JavaScript code into the program that runs on a web page [28]. The server delivers the code for the page, which now contains the injected script, when the victim attempts to view it. The browser used by the end-user is the one that runs the malicious script since it believes it came from a reliable source. The session cookie is automatically sent to the attacker by the browser being used by the victim. The script that is launched has the potential to access cookies, see sensitive information stored in the browser, and even change the text of an HTML page [29].

Theft of cookies – a cookie is a piece of data that is shared by a user session between a website and that user's session. A cookie will always be tracked and stored by the server for user personalization. In this scenario, the attacker prevents the cookies from reaching the server by seizing control of the session ID that is now valid [30]. Cookies can be used to track a user's online activity, such as the websites they visit, the products they view, and the advertisements that they click on. This is typically accomplished using third-party cookies to collect data and establish a profile of the user's actions, which can then be used for personalized advertising

or audience insights. However, users can limit cookie tracking by eliminating their browser history or selecting website tracking.

3.4 Dangers at the Transport Layer

Since it provides an end-to-end link between the two processes that are running over the network, the transport layer is sometimes referred to as the "heart" of the OSI model [31, 32]. It does it by segmenting data and incorporating error-control methods into its operations. Error-control methods in the OSI model ensure reliable data transmission by detecting and correcting errors at the data link layer. They can detect errors using checksum or CRC, correct errors with forward error correction, or request retransmission of the data frame. These methods are necessary to minimize errors and ensure correct data transmission over unreliable communication channels. The connection-oriented and connectionless modes of transmission are the two options available to users at the transport layer. Connection-oriented transmission establishes a dedicated logical connection before data transfer, ensuring reliable delivery of data but with increased overhead. Connectionless transmission, sends data as independent packets without a dedicated connection, providing faster delivery but with less reliability. The mode used depends on the application's specific requirements. Transmission Control Protocol, often known as TCP, and User Datagram Protocol (UDP) are the two protocols that are used the most frequently in the transport layer. Both of these protocols are susceptible to attack by malicious cyber actors. TCP and UDP are both vulnerable to attacks from malicious actors due to errors in their design and the applications that use them. While UDP lacks error checking and flow control, making it easier for attackers to take advantage of application vulnerabilities, TCP is susceptible to SYN flooding, TCP reset attacks, and session hijacking. Attackers can also intercept and modify data packets by using packet sniffing and spoofing. Strong security measures are required to ensure the confidentiality, integrity, and availability of data transmitted over these protocols to minimize these risks. These measures include firewalls, encryption, and intrusion detection and prevention systems.

An example of a DDoS attack is known as a TCP SYN flood. A SYN flood attack takes advantage of an error in the TCP three-way handshake process, causing a server to become overloaded and unable to respond to legitimate requests. This may result in server downtime, resource depletion, financial damage, and increased security risks. Mitigating these attacks necessitates the implementation of appropriate security measures such as firewalls, intrusion detection and prevention systems, and load balancers. The primary objective of this attack is to prevent genuine traffic from accessing the server by using all available resources in a manner that renders the service inaccessible. The SYN flood attack is effective because it takes advantage of the three-way handshake that occurs inside a TCP connection.

1. The attacker uses the faked IP address to send many SYN packets to the server that is the target of the attack. Attackers use IP spoofing, botnets, amplification,

and reflection techniques to send a large volume of SYN packets to a server using a fake IP address. Implementing access control lists and anti-spoofing technologies can help prevent or mitigate these attacks.

2. As a kind of reciprocity, the server will reply to all requests and will ensure that an open port is available to receive the ACK. The server is now waiting for the ACK from the client, but the client never sends it.
3. The server keeps the port open for an extended length of time, which eventually results in the server going down and causing genuine clients to be prevented from receiving service.

The DoS attack known as UDP Flood involves sending a large number of IP packets containing UDP datagrams to the server that is being targeted in order to render it inaccessible. UDP is more susceptible than TCP because, unlike TCP, it does not need an initial handshake to create a connection. TCP and UDP are transport layer protocols used for communication between devices on a network. TCP is connection-oriented, provides reliability, is commonly used for applications that require error-free transmission and has a larger header size. UDP is connectionless, faster, does not provide reliability, is commonly used for applications that require speed and real-time communication and has a smaller header size. The choice between the two protocols depends on the specific requirements of the application being used. The server will employ its resources and look for the applications that are presently listening for the request with a certain port whenever it gets a UDP packet. The port number is supplied in the request. After receiving a large number of UDP packets, all the resources will be used up, which will cause a Denial of Service condition for the traffic that is legal.

3.5 Dangers Posed by the Network Layer

The network layer is the most important part of the OSI model since it is responsible for determining the most efficient logical route that a data packet may take to go from its origin to its destination. The network layer is susceptible to several types of risks, which can compromise network security and stability. Common risks include network intrusion, DoS attacks, routing attacks, ARP attacks, IP spoofing, and malware infections. To prevent potential threats, robust security measures and protocols such as firewalls, encryption, and intrusion detection systems are necessary to safeguard the network. Since the network layer is directly connected to the internet, it is more susceptible to harmful attacks than the lower layers. Attacks like IP spoofing, route poisoning, and denial-of-service attacks can be carried out by attackers by using the advantages of these vulnerabilities. To reduce these risks, network administrators need to put in safety features such as firewalls, intrusion detection and prevention systems, and VPNs, as well as conduct regular security audits to identify and address vulnerabilities.

The process of detecting and monitoring network traffic while simultaneously collecting data packets that are being sent over a network is known as "packet

sniffing". Packet sniffing involves capturing and filtering network traffic to analyze data packets and identify both lawful and malicious activity. The process includes analyzing the contents of each packet and can be used to diagnose network issues, optimize performance, and detect security incidents. However, it is important to use packet sniffing for legitimate purposes only and to implement appropriate security measures to prevent unauthorized use. In order to get access to secret information, attackers will sniff the network in search of sensitive data that is either not encrypted or just partially encrypted and will steal this data. Depending on the accessibility mode, packet sniffing may or may not be considered lawful activity.

Spoofing the identities of data packets is what is meant by the term "packet spoofing". The data that is sent over the network will be divided up into many different packets. Each packet has its own IP header, which includes information about the packet as well as the source IP address and the destination IP address. Hackers make use of the programme to generate IP packets with a changed source address, after which they attempt to interact with the server by masking its IP address and using an IP address that is known to be trustworthy. IP spoofing like this will always result in DDoS assaults. IP spoofing involves falsifying the source IP address in IP packets to impersonate another computer system. Cybercriminals use this technique to carry out malicious activities undetected and bypass security measures that rely on IP blacklisting.

3.6 Dangers Posed by the Data Link Layer

The primary function of the data link layer is to carry out framing, which is the process of attaching the data's source MAC address and its destination MAC address. Also guarantees the flow of data that is free of errors. The data connection layer is composed of two sublayers in its structure:

1. Logical Link Control (LLC),
2. Media Access Control (MAC).

The MAC layer manages access to the physical network and adds a header to data packets, while the LLC layer provides flow control, error checking, and sequencing of frames. Together, these layers control data flow and ensure reliable and efficient transmission between devices on the same network.

There are a few potential dangers that might occur at this layer, including ARP poisoning, MAC flooding, and port theft.

ARP poisoning is a kind of the Man-in-the-Middle attack that enables the attacker to disrupt the communication which is taking place between the devices. ARP is used to convert Internet Protocol (IP) addresses into Media Access Control (MAC) addresses. This is the program's primary purpose. An attacker using ARP poisoning will broadcast bogus ARP packets over a Local Area Network (LAN) in order to connect their MAC address to a valid IP address. Now the attacker is in a position to receive any communications that are sent to valid MAC addresses. As

a consequence of this, the adversary is in a position to be able to intercept, change, or stop the communication directed to the MAC address of the victim.

The next step is known as MAC flooding. In most cases, switches will keep a "MAC table" that is comprised of the MAC addresses of the hosts that are connected to each port on the switch. The attacker will continue to send bogus source MAC addresses to the switch until the table is completely filled. Now the switch is confused, and instead of functioning as a switch, it is acting as a hub by broadcasting the frames to all of the associated hosts.

The term "port stealing" refers to an attack in which the perpetrator takes the traffic that is intended for another section of the ethernet switch. In most cases, switches are able to discover the MAC address and associate it with the appropriate port. Attacks that take use of ethernet switches may be called port theft attacks. The attacker sends a flood of counterfeit frames from a separate port that include the victim's MAC address disguised as the source address. At this point, the data frames that were supposed to be transferred to the genuine port are being sent to the attacker's port instead.

3.7 Dangers Posed by the Physical Layer

The "physical" layer is the one that manages all of the "physical" components. Wires, copper cables, fibre optic cables, routers, endpoints, and sockets are the components that link everything to make a real network. Other components include fibre optic cables. The dangers posed by this layer include the interruption of electric signals that are sent between the network nodes, the cutting of cables in a physical manner, insufficient power, and damages caused by natural catastrophes such as short circuits. In this layer, there is also the potential for man-made disasters on purpose, such as the destruction of the devices, theft, and signal jamming.

4 PROPOSED MODEL

After doing an extensive research on HTTP botnets such as SpyEye, Zeus, Athena, BlackEnergy, and Andromeda, we have discovered that some Simple Network Management Protocol - management information base SNMP-MIB variables undergo substantial transformations whenever an HTTP bot make contacts with its command and control server.

Feature Extraction. The C & C server will receive a significant number of HTTP requests from the HTTP botnet as they are sent automatically. These HTTP requests use valid forms and are sent over standard TCP connections; as a result, it might be difficult to pinpoint where these requests are coming from. In addition, the queries made by bots are often created at random or by repeatedly

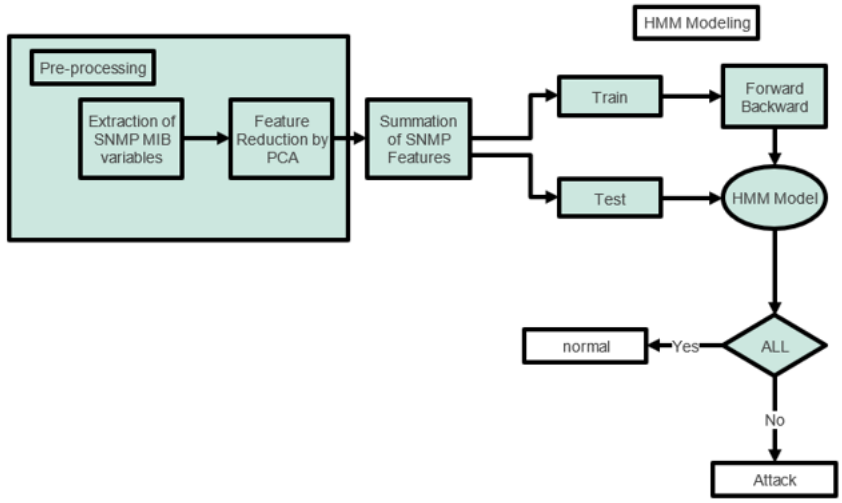


Figure 2. Block diagram of the proposed HsMM model

making a small number of straightforward HTTP requests across TCP connections.

Principal Component Analysis (PCA) is a statistical method that transforms correlated variables into uncorrelated ones called principal components. The components are ranked by importance, with the first accounting for the most variance in the data. PCA is used to simplify data, reduce noise, and identify patterns in machine learning, data analysis, and preprocessing. PCA is performed on the gathered MIB variables, in order to determine which SNMP-MIB variables are important. It is among the dimensionality reduction methods that is commonly quite often utilised for analysis of data and compression. Dimensionality reduction is popular because it simplifies complex data by reducing the number of variables. It improves analysis and visualization, and helps identify patterns and relationships that may not be visible in high-dimensional space. It is useful in fields such as image and natural language processing. It works by identifying a small number of orthogonal linear relationships of the set of variables that had the highest variance in order to reduce a relatively large number of variables into a smaller number of uncorrelated variables. Dimensionality reduction, which involves reducing a high number of variables into a smaller number of uncorrelated ones; it aims to simplify and optimize the analysis of complex datasets while retaining important patterns. It improves machine learning performance, reduces computational costs, and enhances data interpretability and visualization. Additionally, it eliminates redundancy and noise in the data, leading to more accurate analysis. One of the most widely popular methods for dimensionality reduction is this one.

$$\left.\begin{aligned} p_1 &= \mu A_1(x) \\ p_2 &= \mu A_2(x) \end{aligned}\right\} \tag{1}$$

The output response of layer l for feature set 2 is

$$q_j = \mu B_j(y), \tag{2}$$

where

- j represents number of nodes in layer l for feature set 2 and in this research work,
- j varies from 1 to 2 and hence Equation (3) is represented as the output responses of layer 2:

$$w_i = \mu A_i(x) \cdot \mu B_i(y), \tag{3}$$

  where

  – i represents the number of nodes in layer 2 for feature set 1 and in this research work,
  – i varies from 1 to 2 and hence Equation (4) is represented as

$$\left.\begin{aligned} w_1 &= \mu A_1(x) \cdot \mu B_1(y) = p_1 \cdot q_1 \\ w_2 &= \mu A_2(x) \cdot \mu B_2(y) = p_2 \cdot q_2 \end{aligned}\right\} \tag{4}$$

  The output responses of layer 3 are

$$\begin{aligned} \overline{w_l} &= \frac{w_l}{w_1 + w_2}, \\ \overline{w_1} &= \frac{w_1}{w_1 + w_2}, \\ \overline{w_2} &= \frac{w_2}{w_1 + w_2}. \end{aligned} \tag{5}$$

The output response of layer 4 is

$$\left.\begin{aligned} k_i &= \overline{w_l} \cdot f_i \\ k_1 &= \overline{w_1} \cdot f_1 \\ k_2 &= \overline{w_2} \cdot f_2 \end{aligned}\right\} \tag{6}$$

The output response of layer 5 is

$$\begin{aligned} f &= \sum_{i=1}^{2} \overline{w_l} \cdot f_i, \\ f &= \overline{w_1} \cdot f_1 + \overline{w_2} \cdot f_2. \end{aligned} \tag{7}$$

Constructing a HsMM in order to generate a profile of the typical behaviour of MIB traffic is the first step in developing the model that will be used to identify the botnet. MIB traffic analysis can identify botnets by detecting specific network communication patterns between infected devices and their command-and-control servers. The MIB provides a standardized way to monitor and manage network devices, and analyzing MIB traffic patterns can help detect botnet infections. Security analysts can detect and mitigate botnets by monitoring and analyzing MIB traffic. Because this model only takes into account two distinct states, the state space is denoted by the notation $F = 0$ and 1, with 0 denoting the state of the system when it is operating normally and 1 indicating that the system is operating under the direction of the botmaster.

Among the different clustering techniques, the k-means algorithm is one of the major and well-known that uses distance measurements to determine the groups (Euclidean distance metrics). This strategy, among others, is straightforward yet effective. In accordance with the needs of the user, it divides the data into k clusters. The k-means algorithm is a popular and efficient clustering method that outperforms other techniques in terms of scalability, ease of implementation, flexibility, and performance. Its ability to handle large datasets, customize to different applications, and produce accurate results has made it an important tool in data analysis and machine learning. With k-means, data is processed spatially, and after each cluster, the centroid is determined using a statistical technique (mean). Such clustering methods choose random centroids and adjust themselves based on the cluster's circumstances and its divergence (objective function). These algorithms are known as unsupervised because of this. Iterations continue once the procedure starts until the convergence state is reached. The minimization of the distance between the measurement points and the cluster centres is shown by this convergence stage. The number of sub-optimal states in the k-means approach varies depending on the initial value picked.

The k-means' objective function J is given by

$$J = \sum_{i=1}^{m} \sum_{k=1}^{K} w_{ij} \left\| x^i - \mu_k \right\|^2, \tag{8}$$

where μ_k is the k^{th} cluster midpoint, m is number of data points, x^i is the data point in i^{th} vector, K is total number of clusters well-defined by the user, $w_{ij} = 1$ for data point x_i if it fits in to cluster k; else, $w_{ij} = 0$.

The FCM (Fuzzy C-Means) method, which has a benefit over k-means for its ability of detecting mutual clusters, was developed using the k-means technique as a base (i.e., identifying data points that may belong to two or more clusters). FCM is an iterative technique that advances on minimizing the objective function, similar to the k-means approach. The square error analysis is used to determine the convergence process. The convergence process in k-means clustering is evaluated using square error analysis, where the sum of squared distances between data points and assigned cluster centroids is calculated. The k-means algorithm aims to min-

imize the SSE by iteratively updating cluster assignments and centroid positions until convergence. The SSE is used to evaluate the clustering solution and improve its quality. This method is commonly used to assess the convergence process in k-means clustering. This approach activates and starts to cluster utilizing random picks since it follows the essence of unsupervised learning. The three steps of the FCM process are as follows:

1. Cluster Center,
2. Membership function, and
3. Objective function.

The FCM's objective function is provided by

$$J_{\text{iter}} = \sum_{i=1}^{N} \sum_{j=1}^{nc} \left[M^f \left\| D_{x_j - c_j} \right\|^2 \right], \tag{9}$$

where nc is the number of clusters, and N denotes the number of data points in the data vector. (The user may provide this in accordance with their needs.) M^f is a membership function given by

$$M^f \left(\frac{d_{ji}}{d_{ki}} \right) = M_{iter} = \frac{1}{\left[\sum_{k=1}^{nc} \left(\frac{d_{ji}}{d_{ki}} \right)^{\frac{1}{f-1}} \right]}, \tag{10}$$

where $\|D_{x_i} - C_j\|$ and $\left\|D_{x_j} - C_k\right\| - f$ is a fuzzification factor, and d_{ji} and d_{ki} are the distance measurements (Euclidean distance technique is employed) among the data point D_{x_i} and the cluster center C_j. C_j is a cluster centre that is often taken from the M^f provided by

$$C_j = \frac{\sum_{j=1}^{N} M^f D_{x_j}}{\sum_{i=1}^{N} M^f}. \tag{11}$$

The Fuzzy C-Means (FCM) and k-means clustering algorithms differ in their approach to cluster membership, centroid calculation, sensitivity to outliers, computational complexity, and tuning parameters. While FCM allows data points to belong to multiple clusters based on membership degrees, k-means assigns each point to a single cluster; meantime, FCM is less sensitive to outliers and more complex than k-means.

The FCM procedure is as described below.

1. The parameters nc, f and condition of the convergence state S_c are defined in order to initiate the FCM process.
2. Because the approach is unsupervised, the FCM starts by choosing a random cluster centre M^f_{rand} or a random membership function C_j. In this instance, M^f_{rand} is picked for additional processing.

3. The current M^f's Equation (11) is used to determine C_j.
4. At this point, the objective function J_{iter} is determined utilizing Equation (11) and its convergence is examined as described below.

$$J_{\text{iter}} - J_{\text{iter}-1} < S_c, \tag{12}$$

   i.e., the difference among the value produced in the current iteration's goal function and the prior iterations is measured after every iteration. The S_c is used to validate this difference.
5. The loop is terminated and the current M_f is taken into consideration as the output if the convergence state has been reached; otherwise, the current M_f is replaced with the new M_f (i.e., M^f_{new} derived utilizing Equation (12) based on the current C_j). Up till the goal is accomplished, the iteration is performed.

Even while FCM yields superior outcomes, the constraint brought on by local optima makes it more difficult to achieve the convergence state and, occasionally, it never does. This is the basis for the introduction of the fuzzification factor f.

Numerous algorithms were developed based on FCM to get around the drawbacks.

$$J_{\text{iter}} = \sum_{i=1}^{N}\sum_{j=1}^{nc}\left[M^f \left\|D_{x_i} - C_j\right\|^2\right] + \frac{a}{N_R}\sum_{i=1}^{N}\sum_{j=1}^{nc}\left[M^f \left\|D_{x_r} - C_j\right\|^2\right], r \in N_i, \tag{13}$$

where

- a is a constant that is employed for boosting or adjusting and
- N_R is the cardinality.

Let the collection of recognizable symbols, that are nothing more than the summation of SNMP-MIB variables at various time intervals, be denoted by the notation V = "v0, v1, v2ams XM-1". The visible symbol distribution is denoted by the equation $B = bi(k)$, where $i > F$ and $k > M$

The distribution at the beginning of the state. P is equal to $[P0, P1]$, and it is a given that $P0 = 1$ and $P1 = 0$ due to the fact that the model was developed for a perfect normal process at the beginning.

5 TRAINING PHASE

Following the construction of the model, we will need to train the model so that we can determine the parameters of the HsMM. In order to keep the most accurate parameters of a legal HsMM model, the typical forward-backward training procedure is utilised. The forward-backward training procedure, or backpropagation algorithm, is a method used to train neural networks. It involves propagating input data forward through the network, computing the output, calculating the error between the

actual and desired output, backpropagating the error through the network to update neuron weights, and repeating the process iteratively until the error is minimized or a stopping criterion is met. This procedure helps the network learn from errors and improve its accuracy. The parameters of the HsMM are first calculated, and then they are re-evaluated in such a way that the ALL function gradually grows until it reaches its highest possible value. ALL may be calculated as follows, assuming that we have the sequence O1: T and the model: parameter.

$$M^f = \frac{\left(\left\|D_{x_i} - C_j\right\|^2 \frac{a}{N_n} \sum_{r \in N_i} \left\|D_{x_r} - C_j\right\|^2\right)^{\frac{1}{f-1}}}{\sum_{M=1}^{c} \left(\left\|D_{x_i} - C_j\right\|^2 \frac{a}{N_R} \sum_{r \in N_i} \left\|D_{x_r} - C_j\right\|^2\right)^{\frac{1}{f-1}}}, \tag{14}$$

$$C_j = \frac{\sum_{i=1}^{N} M^f \left(D_{x_i} + \frac{a}{N_n} \sum r \in N_i D_{x_r}\right)}{(1+a) \sum_{i=1}^{N} M^f}. \tag{15}$$

Equations (14), (15), and (16) provide the cluster centre membership function, and goal function for the FCM_S variant. The settings of FCM_S are changed to create new versions of the algorithm, such as FCM_{S1} and FCM_{S2}.

$$\varepsilon_i = \frac{1}{1+a} \left(D_{x_j} + \frac{a}{N_R} \sum_{j \in N_i} x_j\right). \tag{16}$$

In order to create an EnFCM (Enhanced FCM), the D_{x_i} is further substituted with ε_i where ε_i lessens the clustering process's temporal complexity, where the i^{th} data matrix value is taken into account as the centre of the local window (p_i, q_i), and the j^{th} data matrix value is taken into account as the neighbour of the centre. For efficient operation, the scaling factors (λ_s and λ_g) are introduced in this work.

$$\sigma_i = \sqrt{\frac{\sum_{j \in N_i} \left\|x_i - x_j\right\|^2}{N_R}},$$

$$\varepsilon_i = \frac{\sum_{j \in N_i} s_{ij} x_j}{\sum_{j \in N_i} s_{ij}}, \tag{17}$$

where S_{ij} indicates how equivalent a data point is to its neighbour in order to identify the gradient within it. After you have obtained the parameters, you can use HsMM to assess if the supplied observation sequences of SNMP-MIB variables belong to a normal profile or a bot profile by calculating their respective average log likelihoods. This may be done after you have obtained the parameters. The forward-backward technique is used in order to compute the probability that a given sequence would match a particular model. COMPUTATION is performed on EVERY value of the training sequences for BOTH the standard system and the botnet system. In the event if the data from all of the observations

If the figure in question is found to fall within a certain confidence interval, the profile will be regarded as normal. In a similar manner, the confidence interval for botnets is determined. If the whole observation sequence falls within this confidence interval, then the event in question will be categorised as botnet communication.

6 DATASETS USED

The experimental setup allows the deployment of SpyEye, BlackEnergy, Zeus, Athena, and Andromeda botnets. Zeus, ZeuS, or Zbot is a Trojan horse that steals financial information through form snatching and man-in-the-browser keyboard recording. It is commonly used for banking theft and CryptoLocker malware installation. Most Zeus infections occur through drive-by downloads and phishing scams. Since 2011, Zeus malware construction kit's capabilities have been combined into SpyEye, a sophisticated and deadly malware kit. SpyEye, a banking virus, poses global problems and is difficult to identify and eliminate on infected Windows machines. Athena is a C++ DDoS botnet that targets Windows PCs with sophisticated strategies, targeting web, gaming, VoIP, and residential connections.

The Russian hacker underground employs BlackEnergy, a web-based DDoS network, to launch various attacks. The botnet is user-friendly and command-friendly, with minimal syntax and structure. It connects to its controlling servers via HTTP, enabling easy communication between the attackers and the servers. Andromeda, a botnet using the HTTP protocol, is a popular distribution channel for malicious actors. It is only downloaded on 3.8 % of infected computers, with PoS malware being primarily created for stealing credit card data from POS systems. The hackers behind GamaPoS are targeting POS systems among the massive number of compromised PCs, as most machines infected with Andromeda backdoors do not currently run PoS software. The feature extraction component is built using Java, and the SNMP setup tool is installed. MIB variables are gathered from zombie machines for seven and twelve hours daily, with outgoing and incoming traffic contributing to their updating. Table 1 provides information on the datasets.

Botnet MIB Traces			
Botnet	MIB Trace size	Botnet	MIB Trace size
SpyEye	1.25 GB	Athena	2.73 GB
BlackEnergy	2.96 GB	Andromeda	4.59 GB
Zeus	2.57 GB		
Normal MIB Traces			
FTP service	4.95 GB	Web service	4.27 GB
E-mail service	3.28 GB	Remote service	2.90 GB

Table 1. Description of the MIB datasets

The fluctuation of MIB variables seen during regular and botnet communications may be seen in Figures 3 and 4. On the x-axis, each of the figures is shown at a time interval of thirty seconds.

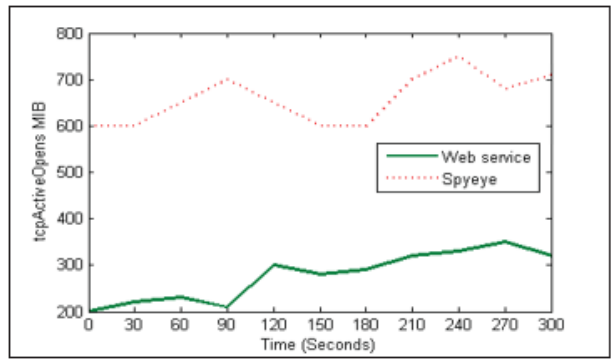


Figure 3. SpyEye and the web service's tcpActiveOpens MIB

Figure 3 demonstrates that there has been a substantial shift in the value of the tcpActiveOpens MIB variable held by the SpyEye botnet in comparison to the value held by the typical web service.

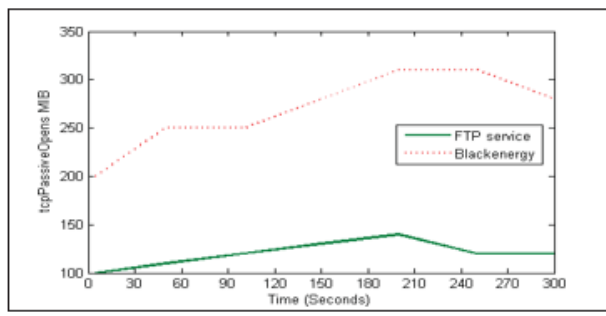


Figure 4. tcpPassiveOpens MIB in FTP service and BlackEnergy

When compared to the values of the standard FTP service's tcpPassiveOpens MIB variable, the tcpPassiveOpens MIB variable values of the BlackEnergy botnet show a substantial difference from those of the normal FTP service. This can be deduced from Figure 4. During the first stages of the bot's spread over the network, the zombie machine attempted to establish a line of communication with a distant site in the hopes of obtaining the address of the C & C server or information from the botmaster. The value of the MIB variable known as tcpPassiveOpens has significantly shifted as a consequence of this behaviour.

Comparison of the tcpCurrEstab MIB variable used by the Email service and the Athena HTTP botnet can be seen in Figure 5. One may deduce, based on the

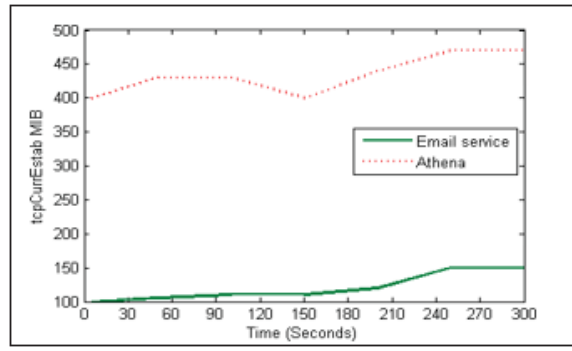


Figure 5. tcpCurrEstab MIB in Email service and Athena

graphic, that the tcpCurrEstab MIB variable undergoes considerable modifications over the course of the Athena HTTP bot propagation.

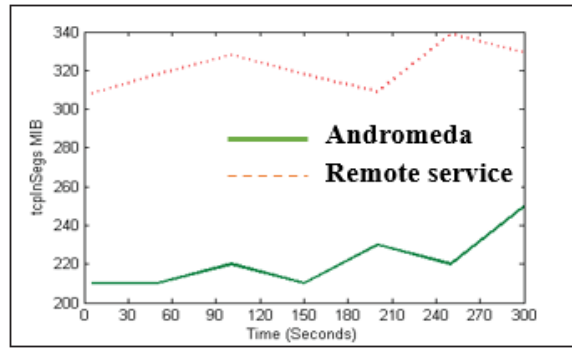


Figure 6. tcpInSegs MIB in remote service and Andromeda

A comparison of the tcpInSegs MIB variable used by Remote service and the Andromeda HTTP botnet can be seen in Figure 6. One can deduce from the figure that there is a significant shift in the value of the tcpInSegs MIB variable that occurs when the Andromeda HTTP bot is in the process of propagating.

Web botnets do not keep a connection with the command and control server, but they do routinely make web requests in order to retrieve the instructions at predetermined intervals. On the other hand, the activities of the botnet are entirely controlled by the computer and have nothing to do with the actions of the users. The observation sequence Ot indicates the summation of chosen SNMP-MIB variables count computed during the t^{th} second in the current state at the host machine while the bot propagation scenario is being played out. In this case, the observation period is thirty minutes, and the summation of the SNMP-MIB variables is computed once every thirty seconds. This is done whenever there are substantial changes in the

present state of those variables. The research and experimental analysis conducted were both point to the duration of thirty seconds as the appropriate reaction time from the botmaster, as shown in Figure 6.

7 EXPERIMENTAL RESULTS

An HsMM model is trained using 70 percent of the datasets from both the normal and botnets after the important SNMP-MIB variables from both the normal and bot propagation systems have been extracted. For instance, we began by instructing the HsMM model using seventy percent of the combined web service and SpyEye dataset. The HsMM model that has been trained is next evaluated using the SpyEye dataset that has not been trained. Experiments are carried out in the same manner with several additional datasets as well. When analysing the suggested model, accuracy serves as the primary statistic that is considered. The findings of the detection accuracy are shown in Table 2. The detection accuracy is rather high, and the rate of false positives is quite low.

After getting the parameters of the HsMM, the next step is to compute the observation sequences for ALL of the regular traffic, followed by the observation sequences for ALL of the botnet traffic. It has been determined that the level of confidence interval for ALL at the 5 % level is $[-2.5, -0]$ when the system is operating normally and EVERYTHING falls inside the range of confidence that spans 5 %. When the system is talking with the C & C server, the values $[-7.5, -4.0]$ are shown.

Datasets	FPR	Detection Accuracy	Results
FTP service	0 %	100 %	Normal
Web service	0 %	100 %	Normal
SpyEye	1.67 %	98.14 %	Botnet
BlackEnergy	1.58 %	98.72 %	Botnet
Zeus	1.75 %	98.02 %	Botnet
Athena	1.29 %	98.94 %	Botnet
Andromeda	1.47 %	98.62 %	Botnet

Table 2. Performance of the proposed HsMM model

Our model was validated using a Windows computer with the settings Intel (R) CoreTM i5 3317U, 1.70 GHz, and the operating system Microsoft Windows 2000. The validation was performed using a variety of genuine botnets as well as standard data. In order to evaluate the effectiveness of the suggested model, we have made use of R, a statistical tool that is freely available online.

R serves as both a programming language and a platform for statistical analysis and visualisation. It supports a wide variety of statistical techniques, including as time-series analysis, classification, clustering, graphical approaches, linear and nonlinear modelling, conventional statistical tests, and more. In addition to that, it has a very flexible design.

7.1 Running Time

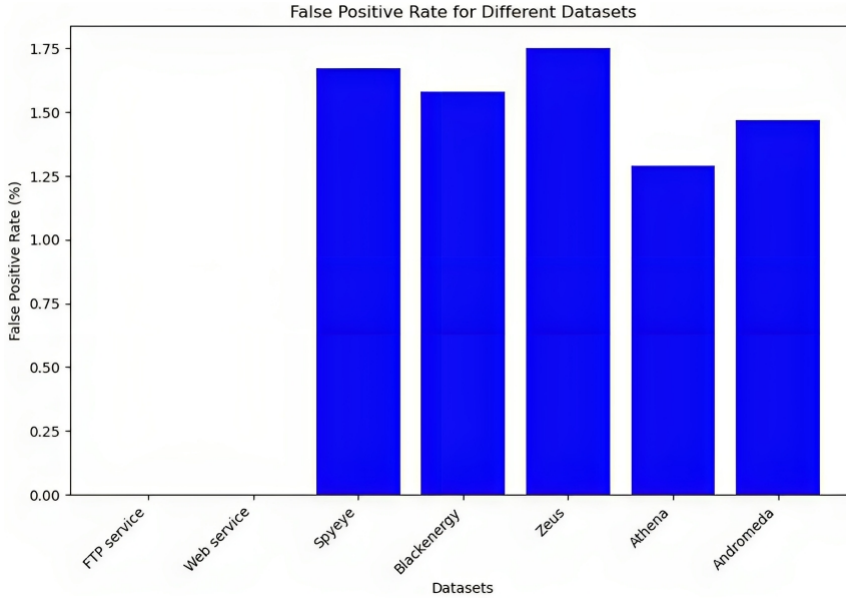
The performance analysis of the suggested model is shown in Figure 7, which presents two important metrics for various datasets: FPR and Detection Accuracy. The fluctuation of the False Positive Rate when the model comes across various datasets is the main topic of Figure 7 a). Conversely, Figure 7 b) presents the model's Detection Accuracy across several datasets. We put the HsMM model through its paces in the system with the settings described earlier so that we could evaluate how well it handled the detection time constraints shown in Figure 8.

It has been noticed that the HsMM model classifies the test instances of web service, FTP service, SpyEye, BlackEnergy, Zeus, Athena, and Andromeda MIB datasets in a span of just 2.05 seconds, 3.10 seconds, 2.51 seconds, 2.95 seconds, 2.73 seconds, 2.84 seconds, and 3.18 seconds, respectively. According to the findings, the HsMM model provides the greatest accuracy while simultaneously reducing the amount of time required for detection.

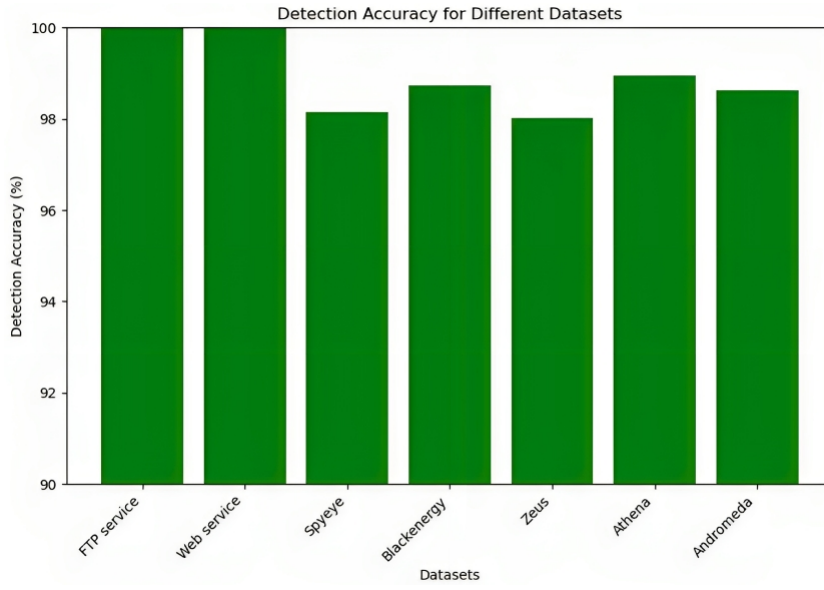
7.2 Analysis in Relation to Previous Works

The accuracy percentages of several approaches for botnet attack detection are shown in Figure 9, with each method's corresponding research publication linked. The accuracy of Naïve Bayes is 97.1 % [33]. Decision Trees have a 99.8 % accuracy rate, according to Alshamkhany et al.'s work on botnet attack detection using machine learning. Achieving a 95 % accuracy rate, Shareena et al. proposed the Deep Neural Network (DNN) technique in their work on an intrusion detection system for IoT botnet assaults using deep learning [34]. Using a deep learning-based intrusion detection system, the BotIDS approach – which was presented by Idrissi et al. in their paper – achieves a high accuracy of 99.94 % in identifying IoT botnet assaults [35]. Using machine learning and the Tree-based Algorithm, Waqas et al. show 99 % accuracy in identifying botnet assaults in Internet of Things devices via a cloud environment [14]. Last but not least, Sriram et al. research on network flow-based IoT botnet attack detection using deep learning describes the K-Nearest Neighbours (KNN) technique, which reaches an accuracy of 99.8 % [36]. These results demonstrate how several machine learning techniques, each with a distinct methodology and accuracy level, can effectively detect botnet assaults.

The M-FLMCM algorithm is an improved image segmentation technique that addresses Fuzzy Local Means' limitations by introducing a modified approach based on local standard deviation and fuzzy c-means clustering. It has shown promising results in medical image segmentation tasks but requires optimal parameter selection and input data quality. Because the M-FLMCM algorithm includes a fuzzy median method, it is one of a kind in that it has the capability to proceeds a median value from a window that contains neighbouring pixels. This enables it to achieve greater precision on usual than other algorithms. The majority of the time, the Zeus botnet were circumscribed by the area that had greater contour (because of the fact that there were not many grayscale fluctuations), as shown in Figure 9.



a) FPR with different datasets



b) Detection Accuracy with different dataset

Figure 7. Performance analysis of the proposed model

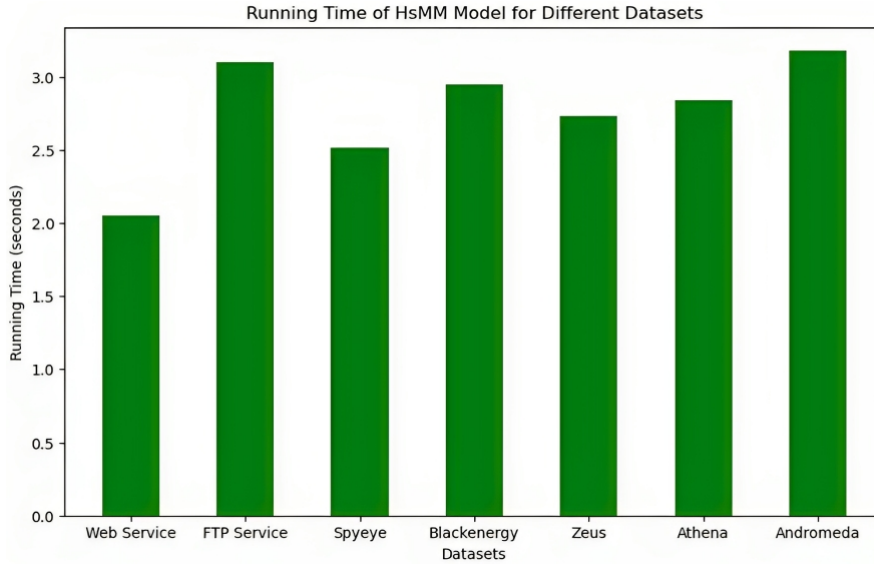


Figure 8. Detection rate

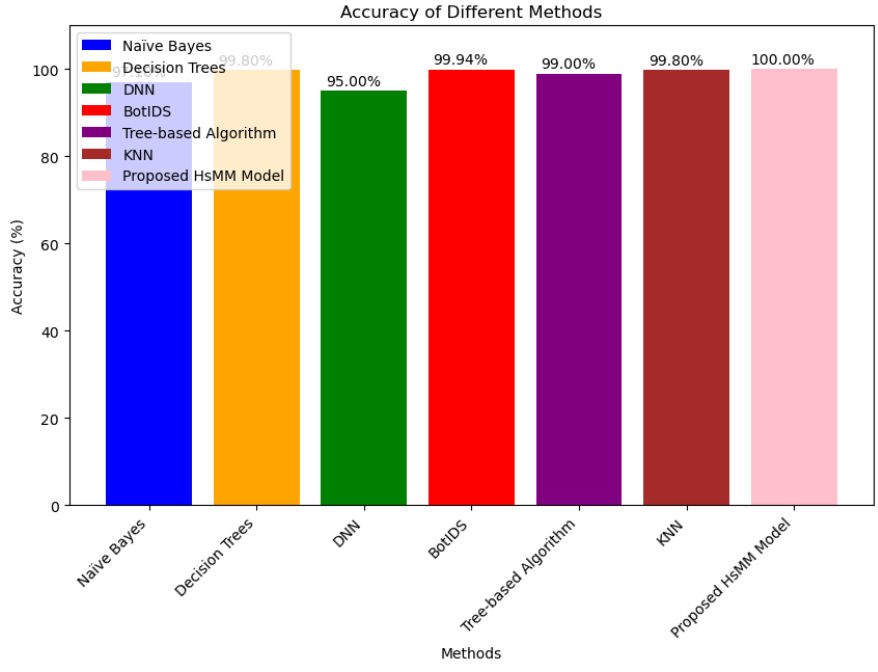


Figure 9. Accuracy comparisons with existing Zeus botnet detection techniques

Figure 10 shows the feature extracted for classifier. Nevertheless, the suggested technique was developed in such a manner that it may successfully construct clusters by identifying even minute changes in the surrounding data neighbourhood. The amount of variation, or noise, in the picture is changed in order to do research on the fundamental aspects of the algorithms. The noise variance, denoted by, provides an estimate of the noise's overall impact on the data. It does this by infecting the image's pixels in order to facilitate the classification process or the clustering of difficult data. The frequencies at which the noises, which have an effect on the data in real time, may range from high to low. In order to conduct this experiment's analysis, the high-frequency noise was collected.

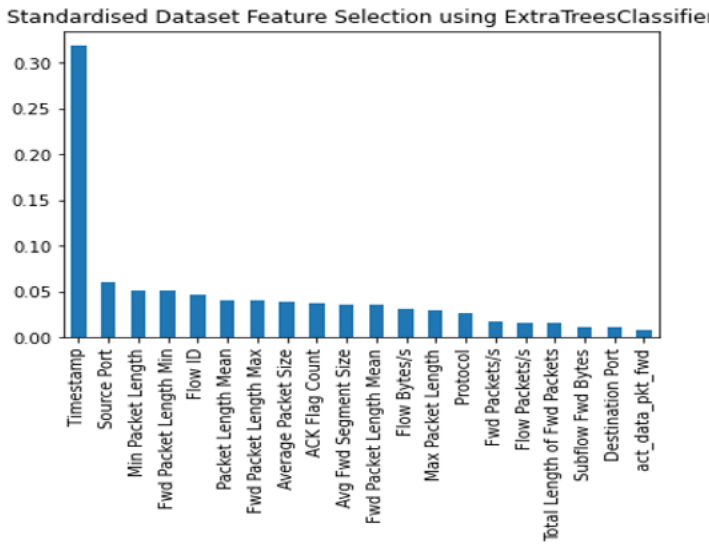


Figure 10. Feature extracted for classifier

In this experiment shown in Figure 11, the amount of noise is changed between four different levels (which can range from 0 to 1), for example, 0.2, which indicates that 20 % of the data is corrupted, 0.4, which indicates that 40 % of the data is corrupted, 0.6, which indicates that 60 % of the information is corrupted, and 0.8, which indicates that 80 % of the data are corrupted. In light of the findings of this validation, M-FLMCM may be considered to have a higher accuracy ratio than various other techniques.

The time-complexity, on the other hand, is the crucial characteristic that has to be regarded equivalent to the accuracy.

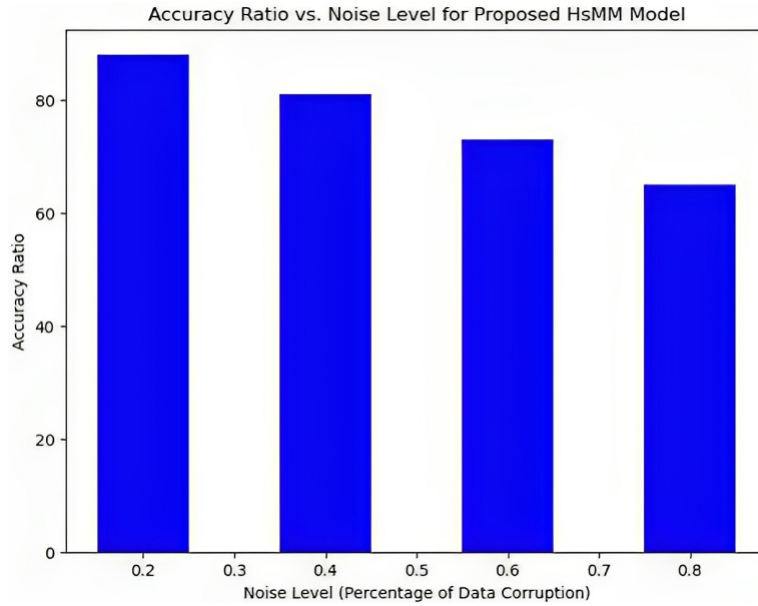


Figure 11. The amount of noise is changed between four different levels

8 CONCLUSION

A host-based botnet identification approach that makes use of TCP-based SNMP-MIB variables as characteristics is presented in this paper. This technique takes into consideration the coordination of HTTP bots within a botnet as well as the malicious behaviour of each individual bot. Utilizing a powerful forward-backward learning technique, these characteristics are plugged into the HsMM in order to get an estimation of the model parameters. The efficiency of the proposed model is shown by its high detection accuracy and low incidence of false positives. Because the suggested technique uses SNMP-MIB variables as features, it is very efficient from a computational standpoint.

The host-based techniques, on the other hand, are used to monitor specific computers to look for suspicious behaviour. This has the capability of detecting attacks while they are in progress or maybe preventing a possible attack from having any impact on the system. This method is not scalable despite the significance of host-based monitoring since it requires the provision of all computers with an extensive collection of efficient monitoring tools. On the other hand, network-based techniques get their data from the network as a whole as opposed to gathering information from each individual host. This article discusses a strategy that is built on networks. The flaws or gaps that exist in software and hardware systems are allowing cyber-attacks to become more sophisticated. The confidentiality, integrity, and credibility of the communication will all be put at risk as a result of the cyber-

threats that have been covered in this article. These difficult challenges to cyber security must be adequately addressed for a variety of preventative measures and corrective steps to be performed.

REFERENCES

[1] Kumar, S.—Dalal, S.—Dixit, V.: The OSI Model: Overview on the Seven Layers of Computer Networks. International Journal of Computer Science and Information Technology Research, Vol. 2, 2014, No. 3, pp. 461–466.

[2] Suresh, P.: Survey on Seven Layered Architecture of OSI Model. International Journal of Research in Computer Applications and Robotics, Vol. 4, 2016, No. 8, pp. 1–10.

[3] Daya, B.: Network Security: History, Importance, and Future. Alpha Wireless, 2013, `http://www.alphawireless.co.za/wp-content/uploads/2013/01/Network-Security-article.pdf`.

[4] Gharibi, W.—Shaabi, M.: Cyber Threats in Social Networking Websites. CoRR, 2012, doi: 10.48550/arXiv.1202.2420.

[5] Sinha, P.—Jha, V. K.—Rai, A. K.—Bhushan, B.: Security Vulnerabilities, Attacks and Countermeasures in Wireless Sensor Networks at Various Layers of OSI Reference Model: A Survey. 2017 International Conference on Signal Processing and Communication (ICSPC), IEEE, 2017, pp. 173–182, doi: 10.1109/CSPC.2017.8305855.

[6] Abomhara, M.—Køien, G. M.: Cyber Security and the Internet of Things: Vulnerabilities, Threats, Intruders and Attacks. Journal of Cyber Security and Mobility, Vol. 4, 2015, No. 1, pp. 65–88, doi: 10.13052/jcsm2245-1439.414.

[7] Messai, M. L.: Classification of Attacks in Wireless Sensor Networks. CoRR, 2014, doi: 10.48550/arXiv.1406.4516.

[8] Obaid, H. S.—Abeed, E. H.: DoS and DDoS Attacks at OSI Layers. International Journal of Multidisciplinary Research and Publications, Vol. 2, 2020, No. 8, pp. 1–9.

[9] Sula, E.: A Review of Network Layer and Transport Layer Attacks on Wireless Networks. International Journal of Modern Engineering Research (IJMER), Vol. 8, 2018, No. 12, pp. 23–27.

[10] Jain, K. M.—Jain, M. V.—Borade, J. L.: A Survey on Man in the Middle Attack. IJSTE – International Journal of Science, Technology and Engineering, Vol. 2, 2016, No. 9, pp. 277–280.

[11] Hasan, T.—Malik, J.—Bibi, I.—Khan, W. U.—Al-Wesabi, F. N.—Dev, K.—Huang, G.: Securing Industrial Internet of Things Against Botnet Attacks Using Hybrid Deep Learning Approach. IEEE Transactions on Network Science and Engineering, Vol. 10, 2023, No. 5, pp. 2952–2963, doi: 10.1109/TNSE.2022.3168533.

[12] Abu Al-Haija, Q.—Al-Dala'ien, M.: ELBA-IoT: An Ensemble Learning Model for Botnet Attack Detection in IoT Networks. Journal of Sensor and Actuator Networks, Vol. 11, 2022, No. 1, Art. No. 18, doi: 10.3390/jsan11010018.

[13] Maurya, S.—Kumar, S.—Garg, U.—Kumar, M.: An Efficient Framework for Detection and Classification of IoT Botnet Traffic. ECS Sensors Plus, Vol. 1, 2022, No. 2, Art. No. 026401, doi: 10.1149/2754-2726/ac7abc.

[14] Waqas, M.—Kumar, K.—Laghari, A. A.—Saeed, U.—Rind, M. M.—Shaikh, A. A.—Hussain, F.—Rai, A.—Qazi, A. Q.: Botnet Attack Detection in Internet of Things Devices over Cloud Environment via Machine Learning. Concurrency and Computation: Practice and Experience, Vol. 34, 2022, No. 4, Art. No. e6662, doi: 10.1002/cpe.6662.

[15] Alazab, M.: A Discrete Time-Varying Greywolf IoT Botnet Detection System. Computer Communications, Vol. 192, 2022, pp. 405–416, doi: 10.1016/j.comcom.2022.06.016.

[16] Syamsuddin, I.—Barukab, O. M.: SUKRY: Suricata IDS with Enhanced kNN Algorithm on Raspberry Pi for Classifying IoT Botnet Attacks. Electronics, Vol. 11, 2022, No. 5, Art. No. 737, doi: 10.3390/electronics11050737.

[17] Almseidin, M.—Alkasassbeh, M.: An Accurate Detection Approach for IoT Botnet Attacks Using Interpolation Reasoning Method. Information, Vol. 13, 2022, No. 6, Art. No. 300, doi: 10.3390/info13060300.

[18] Pan, X.—Yamaguchi, S.—Kageyama, T.—Kamilin, M. H. B.: Machine-Learning-Based White-Hat Worm Launcher in Botnet Defense System. International Journal of Software Science and Computational Intelligence (IJSSCI), Vol. 14, 2022, No. 1, pp. 1–14, doi: 10.4018/IJSSCI.291713.

[19] Ahmed, A. A.—Jabbar, W. A.—Sadiq, A. S.—Patel, H.: Deep Learning-Based Classification Model for Botnet Attack Detection. Journal of Ambient Intelligence and Humanized Computing, Vol. 13, 2022, pp. 3457–3466, doi: 10.1007/s12652-020-01848-9.

[20] Tuan, T. A.—Long, H. V.—Son, L. H.—Kumar, R.—Priyadarshini, I.—Son, N. T. K.: Performance Evaluation of Botnet DDoS Attack Detection Using Machine Learning. Evolutionary Intelligence, Vol. 13, 2020, No. 2, pp. 283–294, doi: 10.1007/s12065-019-00310-w.

[21] Gupta, B.—Agrawal, D. P.—Yamaguchi, S.: Handbook of Research on Modern Cryptographic Solutions for Computer and Cyber Security. IGI Global, 2016.

[22] Moradi, M.—Zulkernine, M.: A Neural Network Based System for Intrusion Detection and Classification of Attacks. Proceedings of the IEEE International Conference on Advances in Intelligent Systems – Theory and Applications, 2004, pp. 15–18.

[23] Shahin, M.—Chen, F. F.—Bouzary, H.—Hosseinzadeh, A.—Rashidifar, R.: A Novel Fully Convolutional Neural Network Approach for Detection and Classification of Attacks on Industrial IoT Devices in Smart Manufacturing Systems. The International Journal of Advanced Manufacturing Technology, Vol. 123, 2022, No. 5-6, pp. 2017–2029, doi: 10.1007/s00170-022-10259-3.

[24] Liu, D.—Hu, W.: Imperceptible Transfer Attack and Defense on 3D Point Cloud Classification. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 45, 2022, No. 4, pp. 4727–4746, doi: 10.1109/TPAMI.2022.3193449.

[25] Kwon, H.—Lee, S.: Ensemble Transfer Attack Targeting Text Classification Systems. Computers and Security, Vol. 117, 2022, Art. No. 102695, doi:

10.1016/j.cose.2022.102695.

[26] Hegazy, H. I.—Tag Eldien, A. S.—Tantawy, M. M.—Fouda, M. M.—TagElDien, H. A.: Real-Time Locational Detection of Stealthy False Data Injection Attack in Smart Grid: Using Multivariate-Based Multi-Label Classification Approach. Energies, Vol. 15, 2022, No. 14, Art. No. 5312, doi: 10.3390/en15145312.

[27] Mukherjee, D.—Chakraborty, S.—Ghosh, S.: Deep Learning-Based Multilabel Classification for Locational Detection of False Data Injection Attack in Smart Grids. Electrical Engineering, Vol. 104, 2022, No. 1, pp. 259–282, doi: 10.1007/s00202-021-01278-6.

[28] Hu, H.—Salcic, Z.—Sun, L.—Dobbie, G.—Yu, P. S.—Zhang, X.: Membership Inference Attacks on Machine Learning: A Survey. ACM Computing Surveys (CSUR), Vol. 54, 2022, No. 11s, Art. No. 235, doi: 10.1145/3523273.

[29] Islam, U.—Muhammad, A.—Mansoor, R.—Hossain, M. S.—Ahmad, I.—Eldin, E. T.—Khan, J. A.—Rehman, A. U.—Shafiq, M.: Detection of Distributed Denial of Service (DDoS) Attacks in IOT Based Monitoring System of Banking Sector Using Machine Learning Models. Sustainability, Vol. 14, 2022, No. 14, Art. No. 8374, doi: 10.3390/su14148374.

[30] Rafiq, H.—Aslam, N.—Ahmed, U.—Lin, J. C. W.: Mitigating Malicious Adversaries Evasion Attacks in Industrial Internet of Things. IEEE Transactions on Industrial Informatics, Vol. 19, 2022, No. 1, pp. 960–968, doi: 10.1109/TII.2022.3189046.

[31] Punitha, P.—Shanthini, J.—Karthik, S.: A Review on Cluster Based Group Adaptive Hybrid Routing Protocol for Mobility in MANET. 2018 International Conference on Soft-Computing and Network Security (ICSNS), 2018, pp. 1–6, doi: 10.1109/ICSNS.2018.8573627.

[32] Kumar, R. L.—Bhavadharini, R. M.—Karthick, S.—Punitha, P.: A Survey on Detecting Packet Dropping Attacks in MANET. International Journal of Applied Engineering Research, Vol. 10, 2015, No. 85, pp. 132–139.

[33] Alshamkhany, M.—Alshamkhany, W.—Mansour, M.—Khan, M.—Dhou, S.—Aloul, F.: Botnet Attack Detection Using Machine Learning. 2020 14th International Conference on Innovations in Information Technology (IIT), IEEE, 2020, pp. 203–208, doi: 10.1109/IIT50501.2020.9299061.

[34] Jithu, P.—Shareena, J.—Ramdas, A.—Haripriya, A. P.: Intrusion Detection System for IOT Botnet Attacks Using Deep Learning. SN Computer Science, Vol. 2, 2021, No. 3, Art. No. 205, doi: 10.1007/s42979-021-00516-9.

[35] Idrissi, I.—Boukabous, M.—Azizi, M.—Moussaoui, O.—El Fadili, H.: Toward a Deep Learning-Based Intrusion Detection System for IoT Against Botnet Attacks. IAES International Journal of Artificial Intelligence (IJ-AI), Vol. 10, 2021, No. 1, pp. 110–120, doi: 10.11591/ijai.v10.i1.pp110-120.

[36] Sriram, S.—Vinayakumar, R.—Alazab, M.—Soman, K. P.: Network Flow Based IoT Botnet Attack Detection Using Deep Learning. IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2020, pp. 189–194, doi: 10.1109/INFOCOMWKSHPS50562.2020.9162668.

Rami Mohammed Baazeem is the Head of the Management Information Systems (MIS) Department, College of Business, University of Jeddah, KSA. He is a member of many national and international committees. Also, he is the Head of the Quality Assurance unit, College of Business, University of Jeddah. He has 15 years of experience in information systems and management. He published many research papers related to e-commerce and data management. He attended many training courses in his specialization. He obtained his Ph.D. from Kingston University, UK and his M.Sc. from Griffith University, Brisbane, Australia. His research interests include data management, IT governance, e-commerce, and business intelligence.

Computing and Informatics, Vol. 42, 2023, 1091–1113, doi: 10.31577/cai_2023_5_1091

ENHANCED DEEP LEARNING-BASED MODEL FOR SENTIMENT ANALYSIS TO IDENTIFY SARCASM APPEARED IN THE NEWS

Isha Gupta

School of Computer Applications
Manav Rachna International Institute of Research and Studies
Faridabad, India

Indranath Chatterjee*

Department of Computer Engineering, Tongmyong University
Busan, South Korea
&
School of Technology, Woxsen University
Telangana, India
e-mail: indranath.cs.du@gmail.com

Neha Gupta

School of Computer Applications
Manav Rachna International Institute of Research and Studies
Faridabad, India

Abstract. In the field of natural language processing (NLP), detecting emotions or sentiments can be a challenging task, and sometimes emotions can be more complex than just positive or negative. However, detecting sarcasm in textual data adds another layer of complexity. Despite this, identifying the underlying sarcasm in the text has become a recent area of interest among NLP researchers. Headlines in newspapers often use sarcasm to engage readers, but readers may have difficulty

* Corresponding author

recognizing it, leading to a misinterpretation of the news and spreading misinformation. As a result, there is an urgent need for technology that can automatically identify sarcasm with high accuracy. Recent studies in this domain have revealed a need for a robust and efficient model. Deep learning approaches have proven to be effective in sarcasm detection. In this work, we propose a novel two-stage model that uses a word-embedding technique to select relevant features followed by an advanced deep-learning architecture to classify sarcasm in news headlines. Our proposed method demonstrates promising results in identifying sarcasm in text with an accuracy rate of approximately 97 %. We have fine-tuned the hyper-parameters to increase the precision level, which enhances the efficacy of our model. Our work provides a significant contribution to the field of NLP by presenting a reliable and effective model for sarcasm detection. The comparison of our model with recent advancements indicates that our approach outperforms them. By using our model, readers can avoid misinterpretations and the spreading of misinformation. Therefore, our work can have a positive impact on society, and we believe that it can inspire future research in the field of sarcasm detection.

Keywords: Word embeddings, sarcasm detection, deep neural network, GRU, Bi-LSTM, fastText, GloVe, news community

1 INTRODUCTION

Sarcasm entails the use of terms that have a different meaning than what one truly means to express, often to offend or irritate someone or merely for pranks. Recently, studies have looked at how sarcasm is used in a range of everyday scenarios, including circumstances where social standards are broken or at work when giving critique or appreciation. When someone uses sarcasm face-to-face, one might identify it by their expressions. Equally or more helpful, the tone of their voice will likely change, too – they may sound more vibrant or carry out several phrases. However, it can indeed be challenging to decipher sarcasm in written form. In written text, there are no facial cues or vocal tones for a better understanding of the text. Sarcasm detection is a specific area of sentiment analysis [1, 2] which is again a sub-field of natural language programming (NLP) [3] study. Here, the goal is to identify the sarcasm rather than to determine if the emotion is positive or negative. This entails determining whether or if the text is ironic (sarcastic sentiment analysis) [4]. Since sarcasm is more complicated for a computer algorithm to recognize than other human emotions, this field has been fraught with difficulties.

Nowadays, social media platforms play a great role in the communication of text. Twitter and other social media sites have offered authors greater tools to help readers express their intentions. Sarcasm in the text changes a sentence's polarity and reduces the precision of a sentiment analysis task. On Twitter and other social media, people employ relatively casual language, which introduces a limited vocabulary. Whereas in news headlines, there are no spelling errors or colloquial

language, as they are prepared by experts adequately. This lessens the sparsity and raises the likelihood of discovering pre-trained embeddings. News portals frequently appear to use sarcasm in their headlines to engage readers. Furthermore, readers frequently have trouble seeing the sarcasm in the headlines, which leads to them having an erroneous impression of the news and spreading that impression to their peers.

Recent studies use deep learning to recognize sarcasm. Algorithms have been developed to detect the existence of sarcasm and harshness in tweets, customer reviews, and discussion boards. The algorithms were fairly efficient in identifying explicit rude language. However, researchers discovered that algorithms require both linguistic and semantic information to be synthesized to effectively detect sarcasm. Many different models have been developed. Studies from the past indicate that deep neural networks and machine learning techniques yield accurate models.

The objective of our paper is to build an advanced deep learning-based model [5, 6] for the detection of sarcasm accurately. Alongside using a word-embedding algorithm, this work employs the use of Long Short-Term Memory (LSTM), Bidirectional LSTM, and Gated Recurrent Unit (GRU), by tuning the hyper-parameters to achieve optimal results. The major contributions of this work lie in finding the efficacy of the proposed two-stage model, using advanced deep learning techniques for sarcasm detection and comparing it with state-of-the-art algorithms. Our algorithm has a wide range of applications in NLP, including consumer research, sentiment analysis, and knowledge classification. Our algorithm has outperformed all previously developed algorithms, achieving a high level of accuracy. The focus of our proposed model is the ability of the LSTM model to capture temporal dependencies and selectively remember and forget information, making it a powerful tool for sarcasm detection. Furthermore, LSTMs are superior to other models in dealing with long-term dependencies.

We have chosen fastText over other word embedding techniques, as it generates better word embeddings for rare words or even words not seen during training. This is an advantage over other word embedding techniques. Our proposed work overcomes the limitations of individual models, such as overfitting and underfitting, which may also account for the outstanding performance of our model.

Our research demonstrates that fastText embedding improves the performance of the multi-layered LSTM model. However, the disadvantage of our model's complex architecture is that it requires more computational time to train than a simple model. On the other hand, the disadvantage of other developed algorithms is that they use traditional word embedding techniques that do not produce high-accuracy results.

Overall, our algorithm is a significant improvement over previous algorithms and offers a powerful solution for NLP applications, particularly for sarcasm detection.

The paper has been discussed in the following sections. The concept of sarcasm detection and its difficulties are covered in Section 1. The literature review was discussed in Section 2. The main ideas of various deep learning algorithms and word embedding techniques, the proposed model to identify sarcasm in the text,

and the experimental results are covered in Section 3. Section 4 deals with the results obtained, discussions, and comparisons with other cutting-edge algorithms. In Section 5, we extensively analyze the proposed approach. Lastly, Section 6 provides a summary of our study and proposes potential directions for future research.

2 LITERATURE REVIEW

There is already a significant amount of research available on sarcasm detection and identification. Sarcasm is frequently employed in social networks and e-commerce platforms, hence failing to recognize it in tasks involving NLP can result in false positives, such as sentiment analysis and opinion mining. According to recent studies, the two language traits of sentiment and incongruity information are helpful for sarcasm identification. Chen et al. [7] suggested a multi-task learning approach that models context incongruity while integrating semantic data and soft sentiment labels to incorporate sentiment cues. The suggested model performs better for the sarcasm detection job with the use of sentiment hints and incongruity information, according to experimental results on datasets.

Băroiu and Trăușan-Matu [8] presented a thorough analysis of the literature on the development of the artificial sarcasm detection task from its inception in 2010 to the present. According to this study, transformer-based architectures and multi-modal techniques have grown in popularity recently. Savini and Caragea [9] reviewed the state-of-the-art and current algorithms and provided robust baselines for sarcasm detection using BERT pre-trained language models. They initially fine-tuned them on relevant intermediate tasks and then investigated a transfer learning system that uses sentiment classification. They have also investigated emotion detection as a separate intermediary task to infuse information into the target task of sarcasm detection, specifically depending on the association between sarcasm and (implied negative) sentiment and general emotions. Trial results on three datasets showed that the BERT-based models outflank numerous past models.

The ability to recognize contradictions is the fundamental issue with sarcasm detection. Ashwitha et al. [10] resolved the issue by compiling a list of target words that vary in interpretation depending on the context and evaluated whether a sentence contains an exact or ironic use of an objective word. Nayel et al. [11] carried out the process using a model based on the support vector machine (SVM). The ArSarcasm-v2 dataset has been used to assess the proposed model.

Abu Farha and Magdy [12] assessed the effectiveness of 24 Arabic-specific models for detecting sarcasm and sentiment in Arabic. The findings demonstrated that the models used more parameters and were trained exclusively on Arabic data performed best. The investigations using AraGPT2 variations demonstrated poor performance when compared to BERT models, suggesting that it might not be appropriate for classification tasks.

Ren et al. [5] suggested a multi-level memory network that incorporates sentiment semantics to store the characteristics of expressions of sarcasm. The first-level

memory network was used to represent the sentiment semantics, and the second-level memory network represented the contrast between the sentiment semantics and the context of each sentence. They also enhanced the memory network in the absence of local information using an upgraded CNN. The experimental outcomes on the Twitter dataset and Internet Argument Corpus (IAC-V1 and IAC-V2). Majumder et al. [13] developed a multitask learning-based framework that analyses the association between sarcasm detection and sentiment analysis using a deep neural network to enhance the performance of both tasks in a multitask learning environment.

Sharma et al. [14] suggested a new autoencoder-based hybrid sentence embedding technique. Their approach employed a bidirectional encoder representation transformer, a universal sentence encoder, and sentence embedding from LSTM-autoencoder. Du et al. proposed a dual-channel CNN [15] that examines the emotional context of the target text in addition to its semantics. SenticNet was used to enhance the LSTM model with common sense. The user's expression patterns were then taken into consideration using the attention method.

Vinoth and Prabhavathy [16] developed an automated sarcasm detection and classification tool for social media utilizing an ASDC-HPTDL model for hyper-parameter-optimized deep learning. To recognize and categorize sarcasm, the attention bidirectional gated recurrent unit (ABiGRU) technique and improved artificial flora algorithm (IAFO) hyper-parameter tuning method were employed. To recognize and comprehend sarcastic behavior and patterns, Goel et al. [6] tried to close the intelligence gap between humans and machines. To identify sarcasm on the internet, the research relied on an ensemble model that combines many neural processing approaches, including LSTM, GRU, and Baseline Convolutional Neural Networks (CNN).

Barhoom et al. [17] developed a model that can recognize sarcasm in headline news using machine and deep learning to investigate how a computer can recognize sarcastic patterns. Razali et al. [18] focused on identifying sarcasm in tweets by fusing hand-crafted contextual data with information retrieved through deep learning. Feature sets were retrieved from CNN architecture and then mixed with meticulously designed feature sets.

Kumar et al. [19] built a feature-rich SVM that outperformed past models by extracting the most important characteristics. To detect sarcastic comments in a given corpus, they presented a multi-head attention-based bidirectional long-short memory (MHA-BiLSTM) network. Son et al. [20] proposed a profound learning model called sAtt-BLSTM convNet that depends on the crossbreed of soft attention-based bidirectional long short-term memory (sAtt-BLSTM) and convolution neural network (convNet), applying GloVe for building semantic word embeddings.

From the vast rigorous literature survey, we concluded that many researchers applied deep neural networks and word embedding for the detection of sarcasm, however, the limitations of the algorithm and efficacy of the approach still lag in terms of domain exchange. To brief, we designed Table 1 to showcase the performance achieved by different authors in their work.

Reference and Dataset	Word Embedding	Techniques used	Performance
SARC [21]	Word2Vec, fastText, GloVe	CNN, LSTM	Acc: 0.82
Sarcasm Corpus 1 [22]	Word2Vec, fastText, and GloVe	Bi-LSTM	Acc: 95.3 %
News Headlines [23]	Word embedding	HA-LSTM	F1: 0.86
SARC [24]	User embedding	CASCADE	F1: 0.86, Acc: 0.79
Real-time mash-up tweets [25]	GloVe	BiLSTM + CNN	Acc: 92.71 %, F-score: 89.05 %
Internet Argument Corpus [5]	GloVe	CNN	F1: 0.87
Eye Movement Database [13]	Sentence-level word representation	GRU with attention	F1-score: 0.87
Tweets [26]	User embedding	CNN	F1-score: 0.74
Tweets [27]	Word2Vec	CNN + Bi-LSTM	F1-score: 0.85
Hotel reservation reviews [28]	GloVe	Proposed model	Acc: 93.85 %
SST2 [29]	fastText	CNN	Acc: 84 %
SemEval 2018 [30]	–	LMTweets + CNN	Acc: 0.883
Tweets [18]	fastText	CNN	Acc: 94 %
SARC dataset [19]	Sentence embedding	MHA-BiLSTM	F1-score: 77.48 %
SemEval 2015 Task 11 [31]	GloVe	Bi-LSTM	Acc: 86.32 %
Semeval 2014 [32]	Word2Vec	CNN	F1: 97.71 %
Tweets [33]	Word embedding	RNN + LSTM	Acc: 91 %
Semeval 2018 Task 3 [34]	–	Google BERT	F1-score: 69.64 %
Ghosh Dataset [35]	Word embedding	Attentive RNN	F1-score: 95.5 %
Reddit Dataset [36]	fastText	BERT-Base-Cased	Acc: 98.25 %
ArSarcasm v2 dataset [37]	Mazajak embeddings	WANLP (AraBERT + CNN-BiLSTM)	Acc: 0.74
MUStARD and sarcasm detection Reddit track [38]	–	Fuzzy logic	Acc: 75.4 %
SemEval 2015 [39]	GloVe	CAT-BiGRU	Acc: 93 %
Rilloff Sarcastic Dataset [40]	Transformer-based contextual embedding	Attention-based BiLSTM	Acc: 93 %

Table 1. Literature showcasing the works on sarcasm detection involving word embedding and AI techniques

3 METHODOLOGY

The proposed methodology for sarcasm detection consists of mainly two phases. In the first phase, feature selection was done by word embedding technique known as fastText and the second phase mainly focuses on classification of sarcasm in news headlines with the help of deep learning techniques. Figure 1 depicts the architecture of the proposed approach.

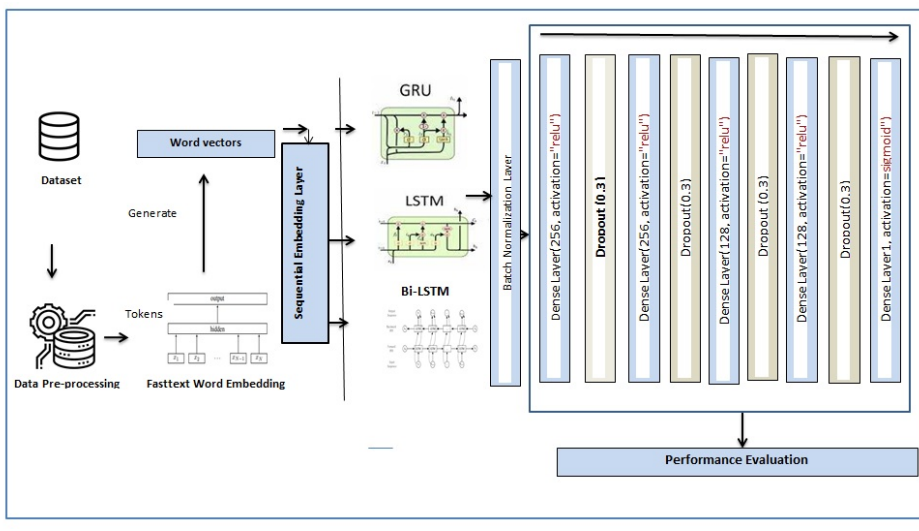


Figure 1. Architectural design of the proposed algorithm

3.1 Word Embedding

Word embedding is a method of portraying words and written materials. It is a lower-dimensional numeric vector input that represents a word, which enables similar representations for words with similar meanings. Word embedding [41] is a technique for taking the features from text and putting them into machine-learning models, which are used for dimension reduction, word prediction, and inter-word semantics detection. While extracting the features, it makes an effort to maintain semantic and syntactical information within the text. The word count in a sentence is used by techniques like BOW, Count Vectorizer, and TF-IDF, although no syntactical or semantic data is saved. The number of vocabulary elements in these methods determines the size of the vector. If the majority of the elements are zero, we can have a sparse matrix. Large input vectors will result in plenty of weights, which will increase the amount of training computation. Word embedding serves as a solution to these issues. The popular word embedding techniques available are Word2Vec by Google, GloVe by Stanford University, and fastText by Facebook.

In this study, we have employed the usage fastText algorithm as a tool for word embedding technique.

3.1.1 FastText

FastText is a library that makes learning word representations and sentence classification quick and easy [42]. It allows multiprocessing while being trained and is written in C++. The user can train supervised and unsupervised word and phrase representations with fastText. Data compression, features in extra models, candidate selection, and fabricators for transfer learning are a few uses for these embedding. Skip-Gram and Continuous Bag of Words (CBOW) are two of the word representation computation models that fastText offers. By using a word that is close by, the Skip-Gram model can learn to anticipate a target word. The CBOW model, on the other hand, makes predictions about the target word based on the context. The target word's fixed-size windows that make up the context are depicted as a bag of words. FastText allows one to train CBOW or Skip-Gram models using softmax, hierarchical softmax, or negative sampling methods.

The distribution of fastText can be defined by the equation as shown in Equations (1) and (2):

$$P(w) = \sqrt{\frac{t}{f(w)}} + \frac{t}{f(w)}, \tag{1}$$

$$f(w) = \frac{count_w}{totalno.oftokens}, \tag{2}$$

where $t = 0.001$ is the chosen threshold, $f(w)$ is the frequency of occurrence for word w.

3.2 Deep Learning Techniques

3.2.1 LSTM (Long Short-Term Memory)

LSTM is a unique class of recurrent neural network (RNN), which is capable of learning long-term dependencies mainly in sequence prediction problems [43]. It can solve the RNN's vanishing gradient problem. LSTM has feedback connections, so it can process all of the data in a sequence.

The architecture of the LSTM model (shown in Figure 2) is different from RNN. In an LSTM model, a memory cell referred to as a "cell state" is vital, as it retains its state over time. Figure 2 shows the cell state as a horizontal line at the top. The cell state can be thought of as a conveyor belt where data passes through without alteration. The gates in LSTMs regulate the addition and removal of information from the cell state. These gates enable data to enter and exit the cell. The mechanism is supported by a sigmoid neural network layer and a point-wise multiplication

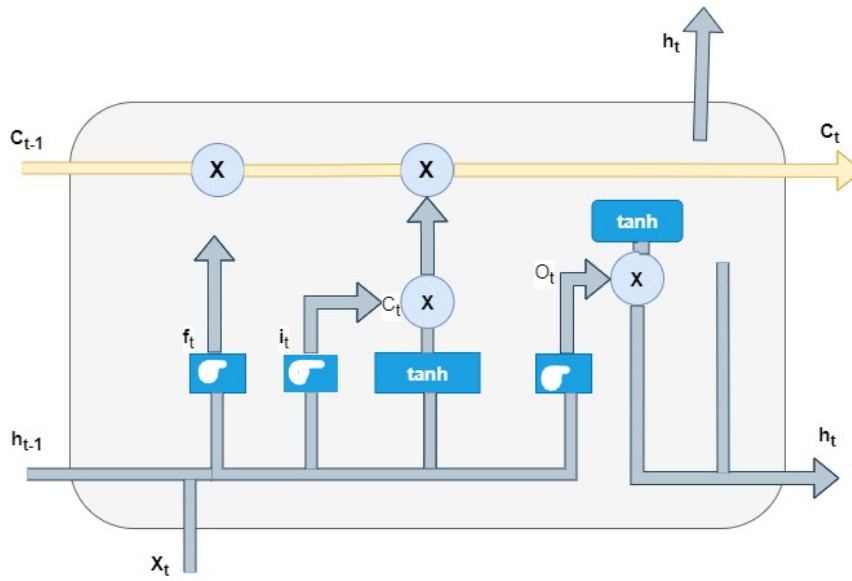


Figure 2. Architecture of LSTM

operation. An LSTM has three gates, to protect and control the cell state. The various equations (as stated in Equations (3), (4), (5), (6), (7) and (8)) associated with LSTM are:

$$f_t = \sigma(W_f.[h_{(t-1)}, x_t] + b_f), \tag{3}$$

$$i_t = \sigma(W_i.[h_{(t-1)}, x_t] + b_i), \tag{4}$$

$$\tilde{C}_t = tanh(W_c.[h_{(t-1)}, x_t] + b_c), \tag{5}$$

$$C_t = f_t * C_{(t-1)} + i_t * \tilde{C}_t, \tag{6}$$

$$O_t = \sigma(W_O.[h_{(t-1)}, x_t] + b_o), \tag{7}$$

$$h_t = O_t * tanh(C_t), \tag{8}$$

where x_t is the input vector, f_t is forget gate activation vector, i_t is input gate activation vector, o_t is output gate activation vector, h_t is the hidden state vector, c_t is the cell state vector.

3.2.2 Bi-LSTM (Bidirectional LSTM)

Bidirectional LSTM resembles an improvement over the generic LSTM algorithm. Each training sequence is presented both forward and backward in bidirectional LSTMs to differentiate recurrent nets. To maintain both past and future informa-

tion, bidirectional input can be made to flow in both ways. The same output layer is shared by both sequences. Bidirectional LSTMs [44] are fully aware of every point in a given sequence, as well as everything that came before and came after it. Speech recognition, text categorization, and forecasting models can all employ this type of network. The structure of Bi-LSTM is shown in Figure 3.

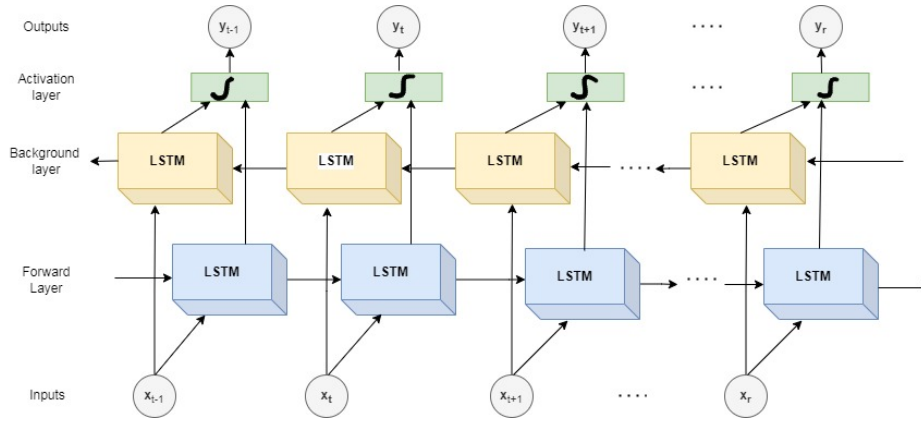


Figure 3. Structure of Bi-LSTM Model

3.2.3 Gated Recurrent Unit (GRU)

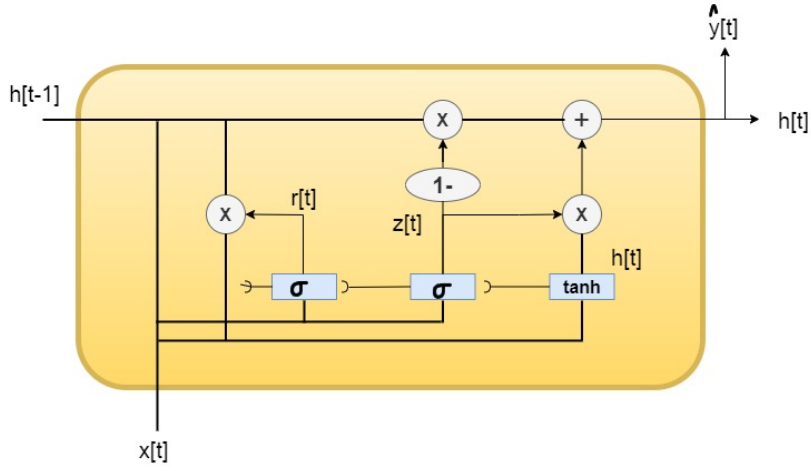


Figure 4. Architecture of GRU

GRU belongs to the RNN family of variations [45]. It only has three gates, unlike LSTM, and it does not keep track of the internal state of the cell. The architecture

of GRU is shown in Figure 4. The data that is kept in the internal cell state of an LSTM recurrent unit is incorporated into the hidden state of the GRU. The next GRU receives this group of data. The internal operation of each recurrent unit in GRU networks, which includes gates that modify the current input and the prior hidden state, is the primary distinction between GRU and RNN. The update gate and reset gate are used to determine the data that should be sent to the output.

The mathematical equations (as shown in Equations (9), (10), (11) and (12)) involved are:

$$z_t = \sigma_g(W_z x_t + U_z h_{(t-1)} + b_z), \tag{9}$$

$$r_t = \sigma_g(W_z x_t + U_r h_{(t-1)} + b_r), \tag{10}$$

$$\tilde{h_t} = \phi_h(W_h x_t + U_h(r_t^{\circ} h_{(t-1)}) + b_h), \tag{11}$$

$$h_t = z_t^{\circ} \tilde{h_t} + (1 - z_t)^{\circ} h_{(t-1)}, \tag{12}$$

where $^{\circ}$ denotes the Hadamard product, x_t and h_t are the input and output vectors respectively, z_t is the update gate vector, r_t reset gate vector, $\tilde{h_t}$ is candidate activation vector, W, U, and b are parameters.

3.3 Experimental Setup

All the experimentations were performed in a High-Performance Computing facility having AMD Ryzen Threadripper PRO 3945 WX processor with 12 cores and 64 GB DDR4 Quad channel RAM. The deep learning architectures were supported with an NVIDIA RTX A5000 graphics card having 24 GB DDR6 memory.

3.3.1 Dataset Details: News Headlines Dataset

The majority of earlier research on sarcasm detection relied on Twitter datasets gathered under hashtag supervision, although these datasets are messy in terms of labels and language. The availability of contextual tweets is also necessary because many tweets are answers to other tweets, making it difficult to discern sarcasm in them. Here, we have used a news headlines dataset [46, 47]. This headlines dataset for sarcasm detection is compiled from two news websites, The Onion, and the HuffPost, to get over the noise constraints in Twitter datasets. Compared to other datasets, these datasets have significant advantages. First headlines are composed with professionalism. There are not any grammatical or spelling errors. Unlike tweets, they carry only text. The only ironic news is printed in The Onion. A total of 28 619 fresh headlines are included in the dataset. The records that fall under the category of sarcastic remarks amount to 13 635. There are 14 984 non-sarcastic records. The collection includes news headline titles as well as labels designating sarcastic and non-sarcastic headlines. The label value "0" indicates that the headline is not sarcastic, whereas the label value "1" indicates that it is. 13 635 records are included in the category of sarcastic remarks. 14 984 records are not ironic.

3.3.2 Data Pre-Processing

At first, the dataset of news headlines has been pre-processed. Raw data were transformed into a format that computers and deep learning algorithms can comprehend and analyze during the data pre-processing stage of the proposed process. The data cleaning stage includes the removal of stopwords, punctuation, and noise.

3.3.3 Application of Word Embedding Technique

The next step was to create word vectors of the pre-processed dataset. This was achieved using the fastText word embedding technique. FastText package facilitates effective word representation and sentence classification learning. As its name implies, it is a quick and effective way to complete the aforesaid task, and because of the nature of its training process, it also ends up learning specific morphological information. FastText is exceptional in generating word vectors for unknown words or from vocabulary words. This is possible because it takes into account the morphological properties of words while generating the word vector for an unknown word. Rare words function well with fastText. To obtain its embeddings and sentence categorization, a word can still be broken down into n-grams, even if it was not observed during training. In our algorithm, around 2 000 000 word vectors are generated from fastText.

3.3.4 Application of Deep Learning Algorithms

In our second phase, the deep neural network (DNN) comes into role. An artificial neural network having more than two layers between the input and output layers is called a deep neural network. The network consists of neurons, synapses, weights, biases, and functions. Keras is a deep-learning API created in Python. One of the layers in Keras is the embedding layer. This is primarily utilized in NLP-related applications and can also be applied to other neural network-based tasks. We utilized Keras embedding layer to train our embeddings. The neurons, synapses, weights, biases, and functions are the components of the embedding layer. So, the first layer of the second phase of our algorithm starts with embedding a sequential model. The value of the embedding dimension was 300.

The input length is the length of input sequences, and it remains constant. In our case, its value was 25. The embeddings_initializer is the embedding matrix which is the output of fastText word embedding. We kept the parameter trainable as 'False'. All of the layer's weights are converted from trainable to non-trainable when layer trainable is set to False. The state of a frozen layer will not be updated throughout training.

The second layer of DNN is set to LSTM, BiLSTM, and GRU one by one. The dimensionality of output space was kept at 256. The next layer is the batch normalization layer. It was used to normalize each mini-batch. As a result, the learning process is stabilized, and the quantity of training epochs needed to train

deep neural networks is dramatically reduced throughout training. There is an addition of a dense layer that is closely coupled to its previous layer, meaning that every neuron in the layer is connected to every other neuron in the preceding layer. The activation function is set to 'ReLu' in the dense layer. To avoid overfitting, the dropout layer of 0.3 (30 %) that randomly sets input units to 0 with a frequency of rate at each step during training. Three more consecutive layers of dense and drop layers are added to the model. Lastly, a dense layer of output unit 1 with the activation function 'Sigmoid' is added to the model. Sigmoid is used for binary classification. Table 2 shows the parameter list exercised in the model.

Parameter Name	**Best Value**
Epochs	50
Batch size	16
Activation function	Sigmoid, ReLu
Embedding dimension	300
Optimizer	Adam
Dropout	0.3
Filter	256, 128

Table 2. List of hyper-parameters and their optimal values of the model

The entire dataset was cross-validated keeping 80 % of the set for training purposes and the rest for the validation set. The hyper-parameters were tuned to find the best value at which our algorithm outperforms other algorithms. The model was evaluated with an epoch ranging from 20 to 200 at an interval of 10. We found that the model reached its optimal capacity at the value of 50 epochs. The value of batch size has been kept in 8, 16, and 32. Two types of activation functions have been used – Sigmoid and ReLu. The embedding dimension has been kept at 300 as the fastText library of 300 dimensions has been used. Adam and Gradient Descent optimizer has been used for the model. Adam has been chosen as it has several advantages that combine the benefits of Gradient with Momentum and RMSProp, such as limited computational requirements, adaptability for non-stationary objectives, and efficient computation with large data and parameters. Different dropout values 0.1, 0.2, and 0.3 has been applied. The best value achieved on the hypertuning of parameters has been shown in Table 2. Here, we have used the following performance metrics to calculate the performance of our model.

3.3.5 Performance Metrics

For calculating the performance of our proposed model, the following metrics (Equations (13), (14) and (15)) are used:

Accuracy: The accuracy of a classifier is calculated by dividing the total number of correctly predicted samples by the total number of samples in the dataset.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}. \tag{13}$$

Precision: Precision refers to the proportion of true positives (TP) to the combined number of TP and false positives (FP).

$$Precision = \frac{TP}{TP + FP}. \tag{14}$$

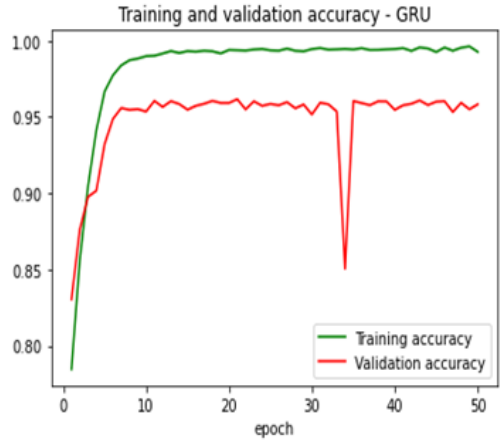
Recall: Recall is the ratio of true positives (TP) by the sum of true positives (TP) and false negatives (FN).

$$Recall = \frac{TP}{TP + FN}. \tag{15}$$

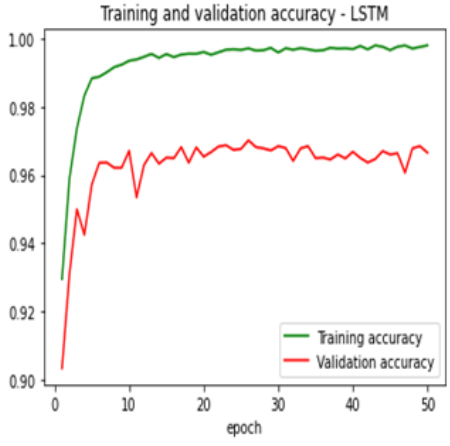
Where TP refers to true positives, TN is true negatives, FN is false negatives and FP is false positives.

4 RESULTS

After the application of the proposed approach incorporating the word-embedding technique and deep learning algorithms, we obtained promising results for all three algorithms, in terms of performance metrics. Figure 5 shows the training v/s validation curves in terms of accuracy results at epoch 50 of LSTM, Bi-LSTM, and GRU. Figure 5 depicts the training and validation accuracy of all three deep learning techniques. This feature highlights one of the advantages of the suggested algorithm. All three techniques achieve a training accuracy of over 99 % and a validation accuracy of over 95 %. Figure 6 displays the performance metrics, including accuracy, precision, and recall of GRU, LSTM, and Bi-LSTM for both training and validation data. The results indicate that the LSTM technique outperforms the Bi-LSTM and GRU models when fastText word embedding is employed in terms of accuracy. Table 3 shows the performance metrics in terms of accuracy, precision, and recall of the proposed algorithm with each of the variants of RNN being used, namely, GRU, LSTM, and Bi-LSTM. We achieved the highest validation accuracy with LSTM, which is 96.70 %. GRU achieved a validation accuracy of 96.06 % and Bi-LSTM achieved a validation accuracy of 95.77 %. The result also indicates the LSTM achieved 99.79 % training accuracy which is the highest value among the other two models. Discussing the validation precision values, GRU has the highest value of 97.71 %, succeeded by a Bi-LSTM with a value of 97.43 %, and LSTM with a value of 97.17 %. Although the highest training precision value was obtained by LSTM, i.e., 99.79 %. LSTM has the highest validation recall value of 97.85 % and training



a) GRU



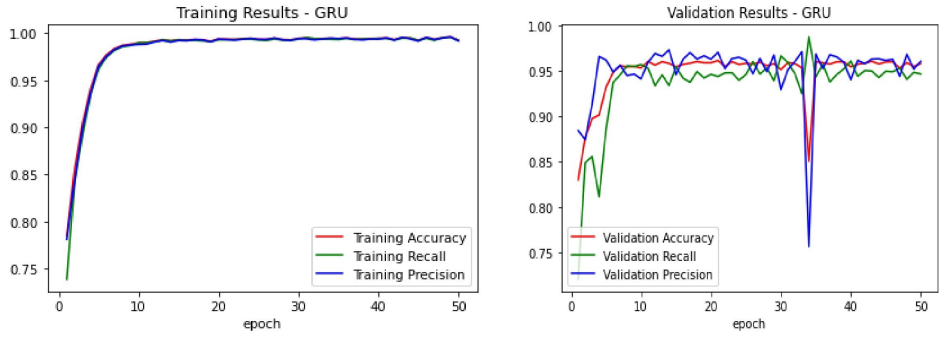
b) LSTM

c) Bi-LSTM

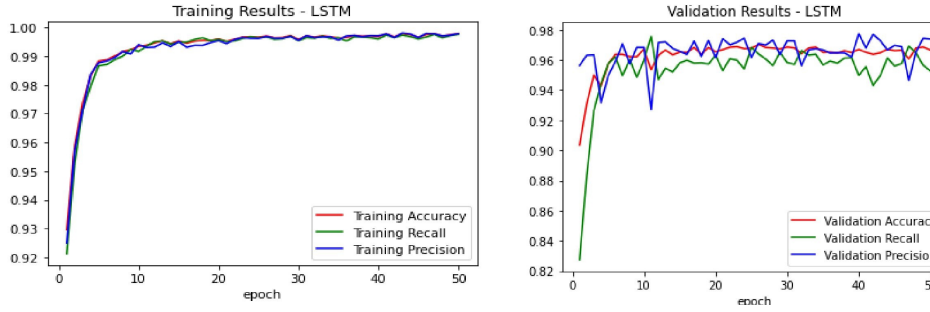
Figure 5. Training accuracy vs validation accuracy for each of the stated algorithms

	Accuracy		**Precision**		**Recall**	
DL technique	**Training**	**Validation**	**Training**	**Validation**	**Training**	**Validation**
GRU	99.28 %	96.06 %	99.20 %	97.71 %	99.23 %	96.08 %
LSTM	99.79 %	96.70 %	99.79 %	97.17 %	99.76 %	97.85 %
Bi-LSTM	99.38 %	95.77 %	99.21 %	97.43 %	99.44 %	96.40 %

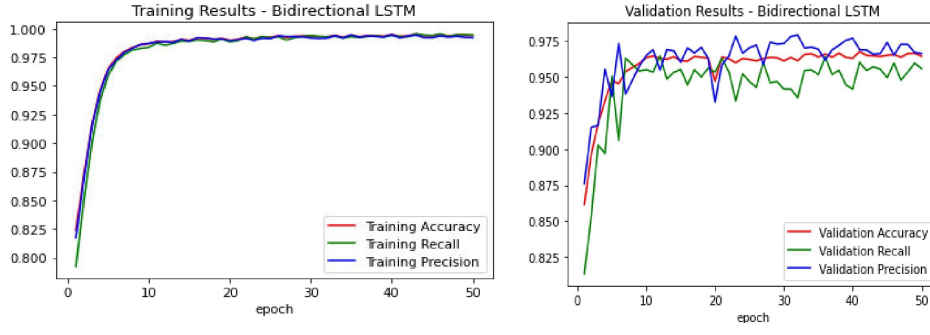
Table 3. Results of the proposed approach



a) Training and validation results for GRU



b) Training and validation results for LSTM



c) Training and validation results for Bi-LSTM

Figure 6. Performance metrics showing the results of training and validation for each of the stated algorithms

Publication	**Word Embedding**	**Deep Learning Model**	**Accuracy**	**Precision**	**Recall**
[6]	GloVe	LSTM	96 %	NA	NA
[36]	fastText	BERT-Base-Cased	92.27 %	92.2 %	92.35 %
[46]	User embeddings	Bi-LSTM + Attention	89.7 %	NA	NA
[48]	Word embeddings	Multi-head attention + GRU	91.6 %	91.9 %	91.8 %
[14]	Sentence embedding	LSTM	90.8 %	92 %	91.2 %
[16]	GloVe	GRU	96.63 %	96.76 %	96.82 %
[17]	GloVe	LSTM	95.27 %	94.15 %	96.62 %
Proposed Model	**fastText**	**LSTM**	**96.70 %**	**97.17 %**	**97.85 %**

Table 4. Comparison results in terms of accuracy with the relevant literature on the news headline dataset

value of 99.76 %. GRU has a 96.08 % recall value whereas Bi-LSTM has a 96.40 % recall value.

Table 4 shows a comparative analysis in terms of the accuracy, precision, and recall of the proposed approach with the recently published relevant literature. All the results were compared on the same dataset, i.e. the news headline dataset. It is observed that our proposed two-stage model comprising fastText word embedding techniques and hyper-parameter-tuned deep learning algorithms has acquired the maximum accuracy of 96.70 %, the precision value of 97.17 %, and recall value of 97.85 % which is the highest among the state-of-the-art algorithms. Our proposed model word embedding technique with fastText with LSTM deep learning technique shows that the technique is very powerful in identifying sarcasm in the News Headlines dataset.

5 DISCUSSION

As discussed so far, we proposed a two-stage model employing the fastText algorithm in the first stage followed by the hyper-parameter-tuned deep learning algorithms, such as GRU, LSTM, and Bi-LSTM. It is evident from the efficacy of our proposed approach that this approach is efficient in detecting sarcasm within the text with a high level of precision. By placing the different models, such as LSTM, Bi-LSTM, and GRU in a deep learning network and varying their hyper-parameters, we attempted to assess the effectiveness of each model. The outcomes showed that the hyper-parameter setup and the dataset we are attempting to examine can sometimes be the cause of performance loss. We have noticed some instances when a significant performance boost has been seen. This corroborates our original hypothesis,

according to which we suggested adjusting the hyper-parameters to learn how the models behave. The results indicate that the LSTM model has performed the best. The LSTM model had been pre-trained using fastText word embedding that enhances the vocabulary of the model and helped in achieving the highest accuracy. In comparison to the past studies, the proposed technique performed noticeably better across all measures. This substantiates our assertion that feature sets explicitly created with fastText word embedding were helpful.

The experiment using more combinations of hyper-parameters, dense, and dropout layers produced positive results. The studies carried out in this study are highly detailed and extensive; they also demonstrate that the best outcomes come from combining several models, which supports our initial hypothesis. In the future, we intend to test the effectiveness of the suggested strategy using a sizable corpus and other domain-specific corpora. Future research will also concentrate on examining previous tweets and user behaviors to better understand their personalities and profiles and, as a result, significantly improve the disambiguation between sarcastic and non-sarcastic language.

6 CONCLUSION

NLP researchers have shown an increasing interest in comprehending textual nature and sarcasm detection, particularly in news headlines. In this paper, we introduce a novel two-stage model that classifies sarcasm in news headlines using GRU, LSTM, and Bi-LSTM and extracts features using a fastText word-embedding technique. Our method yields encouraging outcomes, with a remarkable accuracy rate of 97 %. Our model's hyper-parameters have been further tuned to increase accuracy. Notably, when compared to recent developments in this area, our approach shows higher performance. These results show that our research makes a substantial contribution to the NLP field. Our technique offers practical implications for sentiment analysis and opinion mining in news articles by successfully identifying sarcasm in headlines. Additionally, it raises the standard and dependability of automated text analysis systems. Future research should concentrate on improving our model by investigating new methodologies and using larger datasets. We want to advance the study of sarcasm detection in NLP through ongoing development. Overall, our research highlights the potency of our two-stage model for deciphering ironic text in news headlines and for advancing NLP research.

Acknowledgments

The authors are thankful to the Manav Rachna International Institute of Research and Studies, India, and Tongmyong University, South Korea for providing the platform to implement the research study.

Funding. This research work received no external funding.

Conflict of Interest. The authors declared no conflict of interest.

REFERENCES

[1] Gupta, I.—Chatterjee, I.—Gupta, N.: A Two-Staged NLP-Based Framework for Assessing the Sentiments on Indian Supreme Court Judgments. International Journal of Information Technology, Vol. 15, 2023, No. 4, pp. 2273–2282, doi: 10.1007/s41870-023-01273-z.

[2] Gupta, I.—Chatterjee, I.—Gupta, N.: Latent Semantic Analysis Based Real-World Application of Topic Modeling: A Review Study. 2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS), 2022, pp. 1142–1149, doi: 10.1109/ICAIS53314.2022.9742848.

[3] Chowdhary, K. R.: Natural Language Processing. Fundamentals of Artificial Intelligence, Springer, New Delhi, 2020, pp. 603–649, doi: 10.1007/978-81-322-3972-7_19.

[4] Hussein, D. M. E. D. M.: A Survey on Sentiment Analysis Challenges. Journal of King Saud University - Engineering Sciences, Vol. 30, 2018, No. 4, pp. 330–338, doi: 10.1016/j.jksues.2016.04.002.

[5] Ren, L.—Xu, B.—Lin, H.—Liu, X.—Yang, L.: Sarcasm Detection with Sentiment Semantics Enhanced Multi-Level Memory Network. Neurocomputing, Vol. 401, 2020, pp. 320–326, doi: 10.1016/j.neucom.2020.03.081.

[6] Goel, P.—Jain, R.—Nayyar, A.—Singhal, S.—Srivastava, M.: Sarcasm Detection Using Deep Learning and Ensemble Learning. Multimedia Tools and Applications, Vol. 81, 2022, No. 30, pp. 43229–43252, doi: 10.1007/s11042-022-12930-z.

[7] Chen, W.—Lin, F.—Zhang, X.—Li, G.—Liu, B.: Jointly Learning Sentimental Clues and Context Incongruity for Sarcasm Detection. IEEE Access, Vol. 10, 2022, pp. 48292–48300, doi: 10.1109/ACCESS.2022.3169864.

[8] Băroiu, A. C.—Trăușan-Matu, S.: Automatic Sarcasm Detection: Systematic Literature Review. Information, Vol. 13, 2022, No. 8, Art. No. 399, doi: 10.3390/info13080399.

[9] Savini, E.—Caragea, C.: Intermediate-Task Transfer Learning with BERT for Sarcasm Detection. Mathematics, Vol. 10, 2022, No. 5, Art. No. 844, doi: 10.3390/math10050844.

[10] Ashwitha, A.—Shruthi, G.—R., S. H.—Upadhyaya, M.—Ray, A. P.—Manjunath, T. C.: Sarcasm Detection in Natural Language Processing. Materials Today: Proceedings, Vol. 37, 2020, pp. 3324–3331, doi: 10.1016/j.matpr.2020.09.124.

[11] Nayel, H.—Amer, E.—Allam, A.—Abdallah, H.: Machine Learning-Based Model for Sentiment and Sarcasm Detection. Proceedings of the Sixth Arabic Natural Language Processing Workshop, Association for Computational Linguistics, 2021, pp. 386–389, https://aclanthology.org/2021.wanlp-1.51.

[12] Abu Farha, I.—Magdy, W.: Benchmarking Transformer-Based Language Models for Arabic Sentiment and Sarcasm Detection. Proceedings of the Sixth Arabic Natural Language Processing Workshop, Association for Computational Linguistics, 2021, pp. 21–31, https://aclanthology.org/2021.wanlp-1.3.

[13] Majumder, N.—Poria, S.—Peng, H.—Chhaya, N.—Cambria, E.—Gelbukh, A.: Sentiment and Sarcasm Classification with Multitask Learning. IEEE Intelligent Systems, Vol. 34, 2019, No. 3, pp. 38–43, doi: 10.1109/MIS.2019.2904691.

[14] Sharma, D. K.—Singh, B.—Agarwal, S.—Kim, H.—Sharma, R.: Sarcasm Detection over Social Media Platforms Using Hybrid Auto-Encoder-Based Model. Electronics, Vol. 11, 2022, No. 18, Art. No. 2844, doi: 10.3390/electronics11182844.

[15] Du, Y.—Li, T.—Pathan, M. S.—Teklehaimanot, H. K.—Yang, Z.: An Effective Sarcasm Detection Approach Based on Sentimental Context and Individual Expression Habits. Cognitive Computation, Vol. 14, 2022, No. 1, pp. 78–90, doi: 10.1007/s12559-021-09832-x.

[16] Dakshnamoorthy, V.—Prabhavathy, P.: Automated Sarcasm Detection and Classification Using Hyperparameter Tuned Deep Learning Model for Social Networks. Expert Systems, Vol. 39, 2022, No. 10, Art. No. e13107, doi: 10.1111/exsy.13107.

[17] Barhoom, A.—Abu-Nasser, B. S.—Abu-Naser, S. S.: Sarcasm Detection in Headline News Using Machine and Deep Learning Algorithms. International Journal of Engineering and Information Systems (IJEAIS), Vol. 6, 2022, No. 4, pp. 66–73.

[18] Razali, M. S.—Halin, A. A.—Ye, L.—Doraisamy, S.—Norowi, N. M.: Sarcasm Detection Using Deep Learning with Contextual Features. IEEE Access, Vol. 9, 2021, pp. 68609–68618, doi: 10.1109/ACCESS.2021.3076789.

[19] Kumar, A.—Narapareddy, V. T.—Srikanth, V. A.—Malapati, A.—Neti, L. B. M.: Sarcasm Detection Using Multi-Head Attention Based Bidirectional LSTM. IEEE Access, Vol. 8, 2020, pp. 6388–6397, doi: 10.1109/ACCESS.2019.2963630.

[20] Kumar, A.—Sangwan, S. R.—Arora, A.—Nayyar, A.—Abdel-Basset, M.: Sarcasm Detection Using Soft Attention-Based Bidirectional Long Short-Term Memory Model with Convolution Network. IEEE Access, Vol. 7, 2019, pp. 23319–23328, doi: 10.1109/ACCESS.2019.2899260.

[21] Mehndiratta, P.—Soni, D.: Identification of Sarcasm Using Word Embeddings and Hyperparameters Tuning. Journal of Discrete Mathematical Sciences and Cryptography, Vol. 22, 2019, No. 4, pp. 465–489, doi: 10.1080/09720529.2019.1637152.

[22] Onan, A.—Toçoğlu, M. A.: A Term Weighted Neural Language Model and Stacked Bidirectional LSTM Based Framework for Sarcasm Identification. IEEE Access, Vol. 9, 2021, pp. 7701–7722, doi: 10.1109/ACCESS.2021.3049734.

[23] Pandey, R.—Kumar, A.—Singh, J. P.—Tripathi, S.: Hybrid Attention-Based Long Short-Term Memory Network for Sarcasm Identification. Applied Soft Computing, Vol. 106, 2021, Art. No. 107348, doi: 10.1016/j.asoc.2021.107348.

[24] Hazarika, D.—Poria, S.—Gorantla, S.—Cambria, E.—Zimmermann, R.—Mihalcea, R.: CASCADE: Contextual Sarcasm Detection in Online Discussion Forums. Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018), Association for Computational Linguistics, 2018, pp. 1837–1848, https://aclanthology.org/C18-1156.

[25] Jain, D.—Kumar, A.—Garg, G.: Sarcasm Detection in Mash-Up Language Using Soft-Attention Based Bi-Directional LSTM and Feature-Rich CNN. Applied Soft Computing, Vol. 91, 2020, Art. No. 106198, doi: 10.1016/j.asoc.2020.106198.

[26] Naz, F.—Kamran, M.—Mehmood, W.—Khan, W.—Alkatheiri, M. S.—Alghamdi, A. S.—Alshdadi, A. A.: Automatic Identification of Sarcasm in

Tweets and Customer Reviews. Journal of Intelligent and Fuzzy Systems, Vol. 37, 2019, No. 5, pp. 6815–6828, doi: 10.3233/JIFS-190596.

[27] Ghosh, A.—Veale, T.: Magnets for Sarcasm: Making Sarcasm Detection Timely, Contextual and Very Personal. In: Palmer, M., Hwa, R., Riedel, S. (Eds.): Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, 2017, pp. 482–491, doi: 10.18653/v1/D17-1050.

[28] Onan, A.: Sentiment Analysis on Product Reviews Based on Weighted Word Embeddings and Deep Neural Networks. Concurrency and Computation: Practice and Experience, Vol. 33, 2020, No. 5, Art. No. e5909, doi: 10.1002/cpe.5909.

[29] Khasanah, I. N.: Sentiment Classification Using FastText Embedding and Deep Learning Model. Procedia Computer Science, Vol. 189, 2021, pp. 343–350, doi: 10.1016/j.procs.2021.05.103.

[30] Ahuja, R.—Sharma, S. C.: Transformer-Based Word Embedding with CNN Model to Detect Sarcasm and Irony. Arabian Journal for Science and Engineering, Vol. 47, 2022, No. 8, pp. 9379–9392, doi: 10.1007/s13369-021-06193-3.

[31] Kumar, A.—Garg, G.: Empirical Study of Shallow and Deep Learning Models for Sarcasm Detection Using Context in Benchmark Datasets. Journal of Ambient Intelligence and Humanized Computing, Vol. 14, 2023, No. 5, pp. 5327–5342, doi: 10.1007/s12652-019-01419-7.

[32] Poria, S.—Cambria, E.—Hazarika, D.—Vij, P.: A Deeper Look into Sarcastic Tweets Using Deep Convolutional Neural Networks. CoRR, 2016, doi: 10.48550/arXiv.1610.08815.

[33] Porwal, S.—Ostwal, G.—Phadtare, A.—Pandey, M.—Marathe, M. V.: Sarcasm Detection Using Recurrent Neural Network. 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), 2018, pp. 746–748, doi: 10.1109/ICCONS.2018.8663147.

[34] Shrivastava, M.—Kumar, S.: A Pragmatic and Intelligent Model for Sarcasm Detection in Social Media Text. Technology in Society, Vol. 64, 2021, Art. No. 101489, doi: 10.1016/j.techsoc.2020.101489.

[35] Huang, Y. H.—Huang, H. H.—Chen, H. H.: Irony Detection with Attentive Recurrent Neural Networks. In: Jose, J. M., Hauff, C., Altıngovde, I. S., Song, D., Albakour, D., Watt, S., Tait, J. (Eds.): Advances in Information Retrieval (ECIR 2017). Springer, Cham, Lecture Notes in Computer Science, Vol. 10193, 2017, pp. 534–540, doi: 10.1007/978-3-319-56608-5_45.

[36] Kumar, P.—Sarin, G.: WELMSD – Word Embedding and Language Model Based Sarcasm Detection. Online Information Review, Vol. 46, 2022, No. 7, pp. 1242–1256, doi: 10.1108/OIR-03-2021-0184.

[37] Hengle, A.—Kshirsagar, A.—Desai, S.—Marathe, M.: Combining Context-Free and Contextualized Representations for Arabic Sarcasm Detection and Sentiment Identification. CoRR, 2021, doi: 10.48550/arXiv.2103.05683.

[38] Zhang, Y.—Liu, Y.—Li, Q.—Tiwari, P.—Wang, B.—Li, Y.—Pandey, H. M.—Zhang, P.—Song, D.: CFN: A Complex-Valued Fuzzy Network for Sarcasm Detection in Conversations. IEEE Transactions on Fuzzy

Systems, Vol. 29, 2021, No. 12, pp. 3696–3710, doi: 10.1109/TFUZZ.2021.3072492.

[39] Kamal, A.—Abulaish, M.: CAT-Bigru: Convolution and Attention with Bi-Directional Gated Recurrent Unit for Self-Deprecating Sarcasm Detection. Cognitive Computation, Vol. 14, 2022, No. 1, pp. 91–109, doi: 10.1007/s12559-021-09821-0.

[40] Naseem, U.—Razzak, I.—Eklund, P.—Musial, K.: Towards Improved Deep Contextual Embedding for the Identification of Irony and Sarcasm. 2020 International Joint Conference on Neural Networks (IJCNN), IEEE, 2020, pp. 1–7, doi: 10.1109/IJCNN48605.2020.9207237.

[41] Almeida, F.—Xexéo, G.: Word Embeddings: A Survey. CoRR, 2019, doi: 10.48550/arXiv.1901.09069.

[42] Joulin, A.—Grave, E.—Bojanowski, P.—Douze, M.—Jégou, H.—Mikolov, T.: FastText.zip: Compressing Text Classification Models. CoRR, 2016, doi: 10.48550/arXiv.1612.03651.

[43] 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.

[44] Hameed, Z.—Garcia-Zapirain, B.: Sentiment Classification Using a Single-Layered BiLSTM Model. IEEE Access, Vol. 8, 2020, pp. 73992–74001, doi: 10.1109/ACCESS.2020.2988550.

[45] Dey, R.—Salem, F. M.: Gate-Variants of Gated Recurrent Unit (GRU) Neural Networks. 2017 IEEE 60$^{\text{th}}$ International Midwest Symposium on Circuits and Systems (MWSCAS), 2017, pp. 1597–1600, doi: 10.1109/MWSCAS.2017.8053243.

[46] Misra, R.—Arora, P.: Sarcasm Detection Using Hybrid Neural Network. CoRR, 2019, doi: 10.48550/arXiv.1908.07414.

[47] Misra, R.—Grover, J.: Sculpting Data for ML: The First Act of Machine Learning. 2021.

[48] Akula, R.—Garibay, I.: Interpretable Multi-Head Self-Attention Architecture for Sarcasm Detection in Social Media. Entropy, Vol. 23, 2021, No. 4, Art. No. 394, doi: 10.3390/e23040394.

Isha Gupta is a research scholar at the Manav Rachna International Institute of Research and Studies, Faridabad, India. She has past 11 years of teaching experience at the Vivekananda College, University of Delhi. Her research areas include sentiment analysis, data mining, computer networks, software testing, and recommender systems. She has authored and co-authored research papers in journals and conference proceedings in the area of data mining.

Indranath Chatterjee is working as Professor in the Department of Computer Engineering at Tongmyong University, Busan, South Korea. He is also serving as Adjunct Professor at the School of Technology, Woxsen University, India. He is also involved in the Collaborating Faculty at the International Centre for Neuroscience Research, Tbilisi, Georgia. He received his Ph.D. in computational neuroscience from the Department of Computer Science, University of Delhi, Delhi, India. His research areas include computational neuroscience, schizophrenia, sentiment analysis, medical imaging, and deep learning. He has published 11 books on computer science and neuroscience published by renowned international publishers. To date, he has published numerous research papers in international journals and conferences. He is a recipient of various global awards in neuroscience. He is currently serving as a Chief Section Editor of a few renowned international journals and serving as a member of the Advisory Board and Editorial Board of various international journals and Open-Science organizations worldwide. He is presently working on several projects of government and non-government organizations as PI/co-PI, related to medical imaging and machine learning for a broader societal impact, in collaboration with several universities globally. He is an active professional member of the ACM, OHBM, FENS, ACNM, INCF, etc.

Neha Gupta has done her Ph.D. at the Manav Rachna International University and has a total of 16 years of experience in teaching and research. She is a Life Member of ACM CSTA, Tech Republic and a professional member of IEEE. She has authored and co-authored 70 research papers in SCI/Scopus/peer-reviewed journals and conference proceedings in areas of web content mining, mobile computing, and cloud computing. She has published books with publishers such as Springer, Taylor & Francis, IGI Global and Pacific Book International and has also authored book chapters with Elsevier, Springer, CRC Press, and IGI Global, USA. Her research interests include ICT in rural development, web content mining, cloud computing, data mining, and NoSQL databases. She is an active reviewer for the International Journal of Computer and Information Technology and at various IEEE conferences around the world.

Computing and Informatics, Vol. 42, 2023, 1114–1135, doi: 10.31577/cai_2023_5_1114

PUBLIC OPINION MONITORING AND GUIDANCE ANALYSIS IN THE PROCESS OF NEWS DISSEMINATION FROM THE PERSPECTIVE OF BIG DATA

Wei Shi*, Weiwei Qin

School of Humanities and Law, NanChang HangKong University
NanChang, 330063, China
e-mail: shiweiniat@126.com

Abstract. In order to improve the positive effect of news communication, this paper conducts research on the detection and analysis of public opinion in the process of news communication from the perspective of big data and analyzes the guidance of news communication public opinion. Combined with the actual needs of news dissemination, the numerical accuracy, convergence order, numerical convergence and numerical stability of the SS-CSPH method are mainly analyzed and discussed according to the results of numerical simulation. Moreover, this paper confirms that the smooth function and smooth length do have an impact on the solution of the Strang split-corrected smoothed particle hydrodynamics method (SS-CSPH). In addition, this paper constructs a public opinion monitoring and guidance system in the process of news dissemination. From the simulation evaluation results, it can be seen that the method of public opinion detection and guidance in the process of news dissemination proposed in this paper can play a certain role in the monitoring and guidance of news public opinion.

Keywords: Big data, news dissemination, public opinion monitoring, public opinion guidance

* Corresponding author

1 INTRODUCTION

Inversion news phenomenon is a news phenomenon unique to the rich media era based on the background of Internet technology. The news content is mainly based on social and people's livelihood. As the reporting process goes on, the truth of the core event becomes clearer and clearer, and audience participation and changes in public opinion become part of the news. The phenomenon of reverse news contains an element of distortion, but it is not deliberately fabricated. Reverse news is completely different from fake news. Since the latter has no factual basis or distorts the facts by fiction, its core is false, although it contains factual statements and some detailed descriptions [1]. At the same time, the concept of false news is also different from fake news, which is a "one-sided" or "fragmented" report on a certain news event to be reported. It is factually based, but is not comprehensive and accurate, with the reporter's personal bias and subjective interpretation. According to the simple explanation of the above two concepts, we can actually feel it intuitively: the reports in the "reversed news phenomenon" are not fabricated by journalists [2].

The facts are objective. Due to the reporter's observation angle and thinking set, the report temporarily deviated from the truth of the event. Through the process of amendment and supplementation, the truth of the event is finally presented. Therefore, the "reverse news phenomenon" should not be characterized as "false news" or defined as "false news". The fact distortion in the reverse news phenomenon may be just a temporary blind spot of the journalists' own observation and cognition, and its fundamental connotation is deeper and wider than "false news". Strictly speaking, false news cannot be called "news" in essence, it is more similar to "novels", while false news is news in essence [3]. The phenomenon of reverse journalism includes many processes. From the beginning of the news report to the end, the truth of the whole event is released to the public. It is not only the fact that the news is not true at the beginning of the news report. In fact, in the history of journalism, it is not difficult to find the phenomenon of reverse journalism, but the past is more like a random imagination [4]. With the rapid development of information technology, the phenomenon of reverse journalism is emerging endlessly, and the trend of news normalization is gradually emerging. So, we have to be vigilant. When an accidental phenomenon in the past has become a high-incidence phenomenon now, do we need to examine whether there are problems in today's entire news industry environment and start some targeted research [5]. In the traditional era, the news production mode and technology are relatively simple, and they are often only spread through newspapers and television. At the same time, the production mode of traditional news pays more attention to the authority of its information, and the risk control process is relatively strict. Only through layer-by-layer review can a news information be released, and the manufacturing cycle of news information will be longer [6]. In the new media era, the traditional production mode has been gradually replaced. At present, the news production mode around the network media occupies the C position. An ordinary journalist may only need a few seconds to produce piece of news that is seen by the world. Technological progress has changed the mode of

news communication, and all journalists have stood on the same running line. The news channel is no longer an important factor, and everyone has started to compete for the news itself [7]. The purpose of efficient and streamlined production is to enhance the competitive advantage of the timeliness of news. The rapid and large-scale production has greatly reduced various costs, and the linear decline in costs has led to an exponential increase in quantity. The audience has passively entered an era of information explosion, and the difficulty of selecting effective information has also risen sharply [8].

The advantage of traditional media in the past is content control, but this has become an obstacle to progress in the new media era, and the timeliness can not catch up with the emerging media. So, we can see that most of the news is usually first published in the online media, and then the traditional media begins to follow up. Therefore, in today's media ecology, traditional media, and emerging media account for half of the country, respectively. Only by learning from each other, can traditional media and new media jointly create a healthy and orderly communication environment [9]. Today, the media industry will report news content selectively for its own interests. The reasons for the emergence of selective reporting may exist in several aspects: first, selective reporting is not a derogatory term, it is produced with the emergence of news, and it can even be said that it is the normal situation in news reporting, because there are many news events in the world at any time. If media workers do not screen and distinguish from a professional perspective, this itself is unrealistic. One of the responsibilities of the media is to screen and control information. Therefore, it is the responsibility of the media to select appropriate news for interview and reporting [10]. In today's mixed media ecological environment, media people, on the one hand, play the role of event mouthpiece, on the other hand, play the role of social magnifying glass. Some media people carry out news reporting with coloured glasses and news manufacturing with their own prejudices. Unilateral reporting will do harm to the media industry and society to varying degrees and affect the entire ecological environment [11].

The new media era has subverted the original news communication model. Traditional media have become profitable enterprises, and their market competitiveness is facing challenges. News in the new media era has the characteristics of timeliness and fragmentation. News reports are more focused on real-time reporting, which involves the need to verify news information. Nowadays, the competitive pressure of the news media environment leads to the release of news information without verification. At the same time, some media tend to use seditious headlines or words to mislead the public to pay attention to this news event in order to win users' attention at the first time and gain greater exposure. In such an era of excess information, news materials are never lacking. What is lacking is the short attention of the receiver. Therefore, the first news release that can cause topics has become the key to victory. Among media platforms, news practitioners learned first-hand new news through their own channels, edited, collated and disseminated it in the fastest time, and fermented and disseminated the released news through some grey means, such as "water army", so as to trigger the recipients' forwarding and comments [12].

If the above cannot be achieved, real-time follow-up reporting is also a way to "brush the sense of existence". At the same time, due to various realistic conditions, not all media can interview and confirm the reported events, so journalists will use the comments of netizens as the basis, and their own understanding of the events, so as to have more subjective imagination and description.

The progress of communication technology is changing the way people communicate. The rapid innovation of technology opens a "door to a new world" for people. People can interact directly and in real-time, not limited by regions and borders. In addition, the progress of media technology makes it possible to have a consistent cognition of time in different spaces. Communication between receivers and the original mass communication becomes blurred. Recipients have huge initiative, and the status of past communicators has been weakened. Such a convenient way to obtain information has gradually formed a shortcut like "live broadcast" in people's cognition. Some media and individuals will deliberately magnify a detail or deliberately create false details to achieve their own special purpose [13].

From the perspective of the era of news media transition, in the era of news dissemination, the production and operation process of news is relatively standardized and strict. Reporters themselves have full responsibility and time to verify the source of information, and editorial departments also need to check the accuracy of information. It can be said that such a set of operation processes has greatly reduced the error rate of news releases. However, with the continuous development of information technology, under the influence of today's network communication environment, although traditional media institutions still dominate the development of the overall press, the common influence of diversified communication subjects in the era of financial media has promoted the complexity and variability of network communication in the new media era [14].

From the perspective of big data, this paper conducts research on the detection an analysis of public opinion in the process of news communication, and analyzes the guidance of public opinion in news communication, to improve the positive development of social public opinion.

2 FEATURE RECOGNITION OF NEWS AND PUBLIC OPINION INFORMATION FRAMES

2.1 Split Modified SPH Method for Nonlinear Schrödinger Equation

The research objects considered are mainly the following nonlinear Schrödinger equations without angular momentum rotation terms:

$$i\psi_t(\mathrm{x},t)+\alpha(t)\Delta\psi+V(\mathrm{x},t)\psi+\beta(t)|\psi|^2\psi=0,\quad \mathrm{x}\in \mathrm{i}^d, t\geq 0. \tag{1}$$

Among them, Δ is the Laplace operator, $i=\sqrt{-1}$ is the complex unit, t is the time variable, x the space vector, $\psi(\mathrm{x},t)$ is the complex-valued wave function, $\alpha(t)$ is related to the dispersion efficient, $V(\mathrm{x}, t)$ is the external potential, and $\beta(t)$ is the

nonlinear coefficient. First, we quote the idea of split format to write Equation (1) in the following form:

$$i\frac{\partial}{\partial t}\psi(\mathrm{x},t) = (A+B)\psi(\mathrm{x},t), \mathrm{x} \in \mathrm{i}^d \tag{2}$$

Among them, $A = -\alpha(t)\Delta$ represents a linear operator, and $B = -V(\mathrm{x},t) - \beta(t)|\psi|^2$ represents a nonlinear operator. Then the linear part Equation (3) and the nonlinear part Equation (4) can be solved separately.

For the solution of Equation (4), the conjugate $\overline{\psi(\mathrm{x},t)}$ of $\psi(\mathrm{x},t)$ can be multiplied at both ends getting the following:

$$i\frac{\partial}{\partial t}|\psi(\mathrm{x},t) = A\psi(\mathrm{x},t) = -\alpha(t)\triangle\psi(\mathrm{x},t), \tag{3}$$

$$i\frac{\partial}{\partial t}|\psi(\mathrm{x},t) = B\psi(X,t) = -V(\mathrm{x},t)|\psi(\mathrm{x},t) - \beta(t)|\psi(\mathrm{x},t)|^2\psi(x,t). \tag{4}$$

By conjugating both ends of Equation (5), we get:

$$i\frac{\partial}{\partial t}|\psi(\mathrm{x},t)|^2 = -V(\mathrm{x},t)|\psi(\mathrm{x},t)|^2 - \beta(t)|\psi(\mathrm{x},t)|^4, \tag{5}$$

$$\frac{\partial}{\partial t}|\psi(\mathrm{x},t)|^2 = 0. \tag{6}$$

If $\rho(\mathrm{x},t) = \psi(\mathrm{x},t)|^2$, it can be obtained from (6):

$$\rho(\mathrm{x},t) = \psi(\mathrm{x},t)|^2 \equiv |\psi(\mathrm{x},t_n)|^2 = \rho(t_n,\mathrm{x}), t \geq t_n. \tag{7}$$

Substituting Equation (7) into Equation (4), we can get:

$$i\frac{\partial}{\partial t}\psi(\mathrm{x},t) = -V(\mathrm{x},t)\psi(\mathrm{x},t) - \beta(t)|\psi(\mathrm{x},t_n)|^2\psi(\mathrm{x},t), t > t_n. \tag{8}$$

Equation (8) can be solved directly, and its true solution can be obtained as:

$$\psi(\mathrm{x},t) = e^{i\left(V(x,t)+\rho(t)\psi(x,t_n)^2\right)(t-t_n)}\psi(\mathrm{x},t_n), t \geq t_n. \tag{9}$$

In actual calculation, from time $t = t_n$ to time $t = t_{n+1}$ can usually be calculated by applying the idea of Strang splitting scheme, that is, we can decompose the problem into the following three subproblems to solve:

$$i\frac{\partial}{\partial t}\psi(\mathrm{x},t) = -V(\mathrm{x},t)\psi(\mathrm{x},t) - \beta(t)|\psi(\mathrm{x},t)|^2\psi(\mathrm{x},t) \tag{10}$$

$$i\frac{\partial}{\partial t}\psi(\mathrm{x},t) = -\alpha(t)\Delta\psi(\mathrm{x},t) \tag{11}$$

$$i\frac{\partial}{\partial t}\psi(\mathrm{x},t) = -V(\mathrm{x},t)\psi(\mathrm{x},t) - \beta(t)|\psi(\mathrm{x},t)|^2\psi(\mathrm{x},t) \tag{12}$$

The Strang splitting scheme is second-order accurate and unconditionally stable in the time direction. According to the previous analysis, we can directly obtain the true solutions for the above (10) and (12). For formula (11), the first order symmetric smoothed particle hydrodynamics (SPH) method can be used to solve it. Finally, the following discrete model can be obtained:

$$\psi^{*\mathrm{j}} = e^{i(V(x_j^a,t_n)+\beta(t_n)|\psi_j^n|^2)\triangle t/2}\psi_j^n, \tag{13}$$

$$(q)_j^a = \sum_n \frac{1}{np_k}(\psi_k^* - \psi *_j)\frac{\partial^s W_j k}{\partial X_j^a}, \tag{14}$$

$$\frac{\partial \psi^* *_J}{\partial t} = a(t_n)\sum_k \frac{1}{np_k}(q_k^a - q_j^a)\frac{\partial^s W_j k}{\partial x_j^a}, \tag{15}$$

$$\psi_j^{n+1} = e^{i(V(x_j^a,t_n)+\beta(t_n)|w_j^*|2)\triangle t/2}\psi_j^{**}. \tag{16}$$

Among them, x_j^α represents the α^{th} component of the j^{th} spatial news public opinion information frame, np_k represents the density number corresponding to the k^{th} news public opinion information frame, $\frac{\partial^S W_{jk}}{\partial \mathrm{x}_j^a} = (\mathrm{A}^s)^{-1}\begin{pmatrix}(x_k - x_j)W_j\\(y_k - y_j)W_j\\\mathrm{M}\end{pmatrix}$ represents the corrected kernel gradient, where $\mathrm{A}^S = \left[A_{a\rho}^S\right], A_{a,\rho}^S = \sum_k \frac{1}{np_k}\left(\mathrm{x}_k^\alpha - \mathrm{x}_j^a\right)\left(\mathrm{x}_k^\rho - \mathrm{x}_j^\rho\right)W_{jk}, W_{jk} = W\left(|\mathrm{x}_j - \mathrm{x}_k|, h\right)$ represents the smooth function, h represents the smooth length.

The split correction smoothed particle hydrodynamics (SPH) method is used to solve Equations (10), (11) and (12). When the time step is from t_n to t_{n+1}, the algorithm process for solving the numerical solution ψ_j^{n+1} at the position of space x_j is as follows:

Algorithm 1. The split-modified smoothed particle hydrodynamics (SPH) method algorithm for solving the nonlinear Schrödinger equation is as follows:

1. The algorithm takes ψ_j^n as the initial value, and obtains ψ_j^* by solving Equation (10) as follows:
2. The algorithm takes ψ_j^* as the initial value, and uses Equations (18), (19) and (3) to solve Equation (11) to obtain $\psi_j^{\#*}$:

$$\psi_j^{n+1} = e^{\left\{\left(T(x_{t_n}t_n)+\mu(t_n)\psi_j|'\right)dt/2\right.}\psi_j^{**}.$$

2.2 Numerical Simulation of Nonlinear Schrödinger Equation

The performance of the split correction smoothed particle hydrodynamics (SPH) method in numerical accuracy, numerical convergence and numerical stability under different news and public opinion information frame distributions is studied through six different examples. In order to make the numerical simulation results typical, the

examples with true solutions and the examples without true solutions are selected respectively. The numerical solution obtained by the split finite difference method is compared with the numerical solution of the split modified smoothed particle hydrodynamics (SPH) method to verify the validity and reliability of the split modified smoothed particle hydrodynamics (SPH) method for examples without true solutions.

This section mainly selects the linked list search method to search the news and public opinion information frames in the support domain.

Example 1. Equation (1) takes $\alpha(t) = 1, \beta(t) = 2, V(\mathrm{x}, t) = 0$, namely

$$i\frac{\partial \psi}{\partial t} + \psi_{xx} + 2|\psi|^2\psi = 0, x \in [-20, 20]. \tag{17}$$

The true solution of this equation is:

$$\psi(x,t) = \operatorname{sech}(x - 4t)\exp[i(2x - 3t)].$$

First, we define the following convergence order formula:

$$\text{order} = \log\left(\frac{\text{Error 2}}{\text{Error 1}}\right) / \left(\log\left(\frac{h_2}{h_l}\right)\right). \tag{18}$$

For this example, the quintic spline smooth function is selected for numerical simulation, and the smooth length is selected to represent the frame interval of the initial news public opinion information, and the time step dt = 0.0001 is taken. In the case of uniform distribution of news and public opinion information frames, for example 1, the Strang split-corrected smoothed particle hydrodynamics (SS-CSPH) method can achieve second-order accuracy. In fact, it can be seen that the spatial error and convergence speed of the SS-CSPH method depend on h and $\Delta x/h$, and if $\Delta x/h$ is too large or too small, the numerical format may not converge. Only when the distribution of news and public opinion information frames are uniform and $\Delta x/h$ is a suitable constant, the error order of the discrete format is exactly $O(h^2)$, that is, the corresponding convergence order is 2. In addition, the choice of examples also affects the numerical accuracy of the Strang split-corrected smoothed particle hydrodynamics (SS-CSPH) method in the process of numerical simulation.

Example 2. Equation (1) takes $\alpha(t) = \frac{1}{2}\cos(t), \beta(t) = \frac{\cos(t)}{\sin(t)+3}, V(\mathrm{x}, t) = 0$, namely

$$i\frac{\partial \psi}{\partial t} + \frac{1}{2}\cos(t)\psi_{xx} + \frac{\cos(t)}{\sin(t)+3}|\psi|^2\psi = 0, x \in [-2, 2]. \tag{19}$$

The true solution of this equation is:

$$\psi(x,t) = \frac{1}{\sqrt{\sin(t)+3}}\operatorname{sech}\left(\frac{x}{\sin(t)+3}\right)\exp\left(\frac{i\,(x^2-1)}{2(\sin(t)+3)}\right).$$

In the simulation, for the uniform distribution of news and public opinion information frames, 81 news and public opinion information frames are taken, the corresponding news and public opinion information frame spacing and the time step is dt = 0.0001. For the uneven distribution of news and public opinion information frames, 81 news and public opinion information frames are taken, and the time step of the corresponding news and public opinion information frame spacing distribution is dt = 0.0001.

For the case of uniform distribution, we take the smooth length $h = d_0$. However, for the nonuniform distribution of news and public opinion information frames, we take the smooth length $h_0 = d_0, h_i = (d_{i-1} + d_i)/2\ (0 < i < N-1), h_{N-1} = d_{N-2}$. Among them, h_i represents the smooth length corresponding to the i^{th} news public opinion information frame, d_i represents the i^{th} news public opinion information frame spacing, and N represents the number of news public opinion information frames. This study will use the quintic spline smoothing function for numerical simulation.

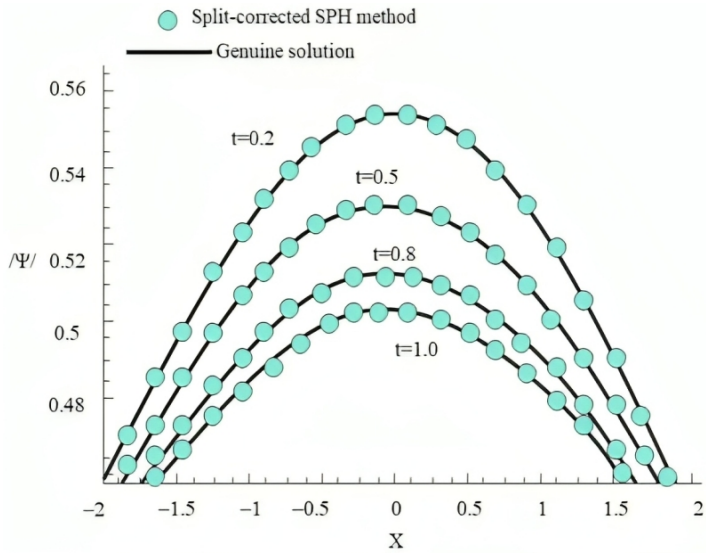


Figure 1. Comparison curve between the numerical solution and the real solution at different times when the distribution of news and public opinion information frames is nonuniform

Figure 1 shows the comparison curves of the numerical solution and the true solution at several different times when the SS-CSPH method has uneven distribution of news and public opinion information frames. It can be seen that in the case of uneven distribution of news and public opinion information frames, the numerical solution obtained by SS-CSPH is in good agreement with the real solution. When the news and public opinion information frames are evenly distributed, the numer-

ical simulation of this example by the SS-CSPH method does not always converge to the second order. In fact, in the process of actually solving the problem, the accuracy, stability and convergence of the SS-CSPH method are often affected by the coefficients of the example, the solution area, as well as the kernel function, the smooth length, and the position of the news and public opinion information frame. Therefore, the theoretical convergence analysis cannot be applied to all cases.

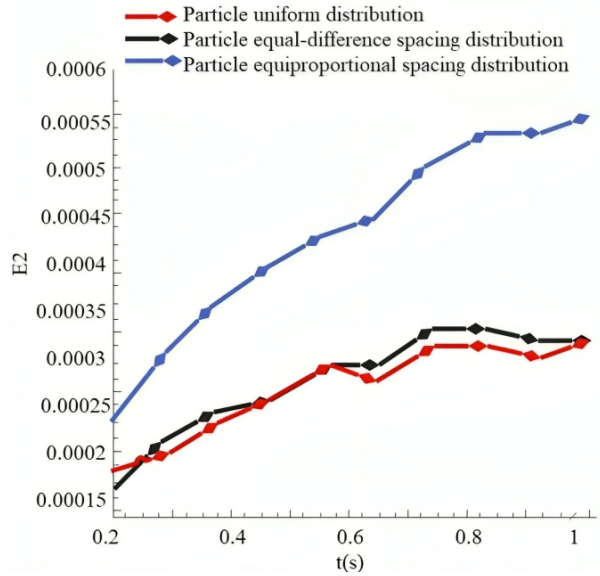


Figure 2. Comparison of the 2-norm error between the numerical solution and the true solution over time under three different news and public opinion information frame distributions

Figure 2 shows the comparison of the 2-norm error between the numerical solution and the true solution at different times when the news and public opinion information frames are uniformly distributed, the news and public opinion information frames are equally spaced, and the news and public opinion information frames are equally spaced. Among them, the 2-norm error is $E_2 = \sqrt{\sum_{i=1}^{N} (\psi_{s,i} - \psi_{a,i})^2} / \sqrt{\sum_{i=1}^{N} \psi_{a,i}^2}$, where $\psi_{s,i}, \psi_{a,i}$ is the numerical solution and the true solution, respectively. It can be seen from Figure 2 that the 2-norm error in the case of non-uniform distribution of news public opinion information frames (equidistant spacing distribution and equal spacing distribution) can achieve the same accuracy as the 2-norm error of uniform distribution of news public opinion information frames. This just shows that the Strang split-corrected smoothed particle hydrodynamics (SS-CSPH) method has a good effect on the nonuniform distribution of news and public opinion information frames. In the above analysis, we use the quintic spline smooth function, and the selection of the smooth function affects the accuracy and stability

of the numerical solution of the smoothed particle hydrodynamics (SPH) method. Therefore, based on this equation, we will study the effect of the smooth function in the numerical simulation of the Strang split-corrected smoothed particle hydrodynamics (SS-CSPH) method when the news public opinion information frames are uniformly distributed and the news public opinion information frames are not uniformly distributed.

It can be seen from Figure 3 that no matter whether the distribution of news public opinion information frames is uniform or not, the error between the numerical solution obtained by using the cubic spline smooth function and the true solution is the smallest. That is, the cubic spline smooth function has the highest accuracy for the numerical solution of Strang split-corrected smoothed particle hydrodynamics (SS-CSPH). However, for the quintic spline smooth function and Gaussian smooth function, the accuracy of the quintic spline smooth function is slightly better than that of the Gaussian smooth function. The absolute error between the numerical solution and the true solution at the time $t = 1$ using three different kernel functions is shown in Figure 4 when the news and public opinion information frames are evenly distributed.

Considering Equation (19), the solution area is $[20, -20]$, the number of news and public opinion information frames are 801, and the time step is unchanged. Figure 5 shows the absolute error between the numerical solution and the true solution using three different kernel functions at the time $t = 1$ under the condition of uniform distribution of news and public opinion information frames. It can be seen that the distribution law of the absolute error value is roughly the same, and there are large oscillations at both ends of the solution area, and the oscillation area of the cubic spline smooth function is the largest, that is, the stability of the cubic spline smooth function is poor.

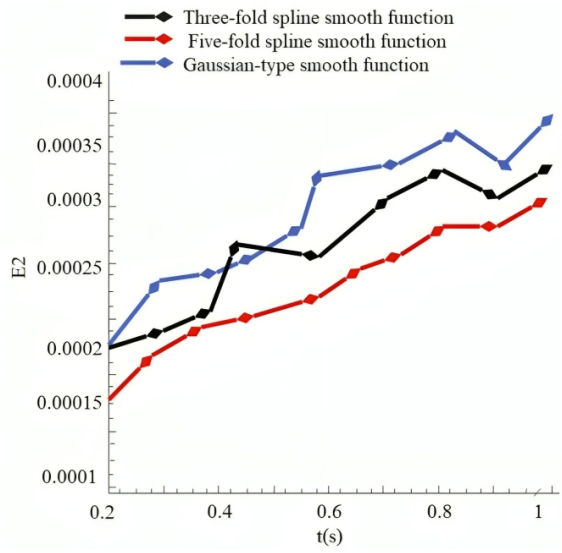
We take the smoothing length $h = d_0$. Figures 5a), b) and c) show the variation of the 2-norm error of the three smoothing functions with different smoothing lengths when the news and public opinion information frames are uniformly distributed. It can be seen that the smoothing length has a significant effect on the effect of the smoothing function in the numerical simulation. For different smooth functions, a certain smooth length can be selected to achieve the best effect in numerical simulation. Conversely, for a fixed smooth length, a smooth function can also be selected to achieve the best effect in numerical simulation.

Example 3. Equation (1) takes $\alpha(t) = \frac{1}{2}, \beta(t) = 1, V(\mathrm{x}, t) = -x\sin(t)$, namely

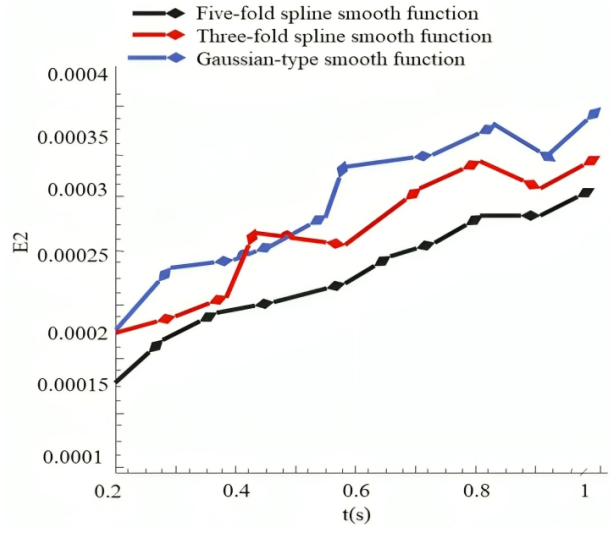
$$i\frac{\partial \psi}{\partial t} + \frac{1}{2}\frac{\partial^2 \psi}{\partial x^2} + |\psi|^2\psi - x\sin(t)\psi = 0, x \in [-20, 20]. \tag{20}$$

The true solution of this equation is:

$$\psi(x,t) = -\frac{3 - 4x^2 + 8x\sin(t) - 4\sin^2(t) - 4t^2 + 8it}{1 + 4x^2 - 8x\sin(t) + 4\sin^2(t) + 4t^2} \times e^{-(i/4)[-4x\cos(t)+\cos(t)\sin(t)-3t]}.$$



a) The comparison chart of the 2-norm error of the numerical solution and the true solution using three different kernel functions under the condition of uniform distribution of news and public opinion information frames over time



b) The comparison chart of the 2-norm error of the numerical solution using three different kernel functions and the true solution under the uneven distribution of news and public opinion information frames with time

Figure 3. Comparison of numerical solutions and true solutions using three different kernel functions

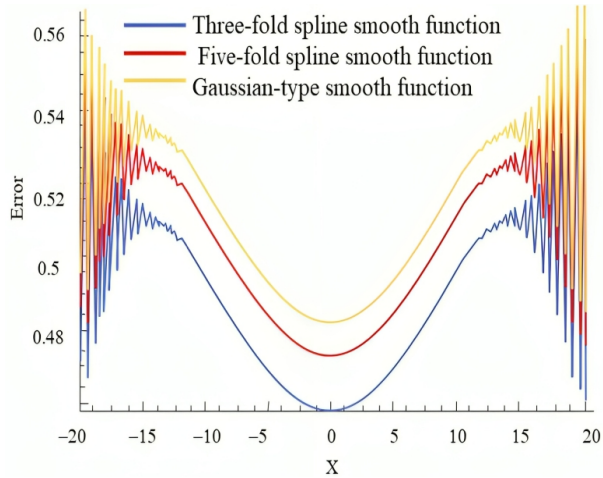
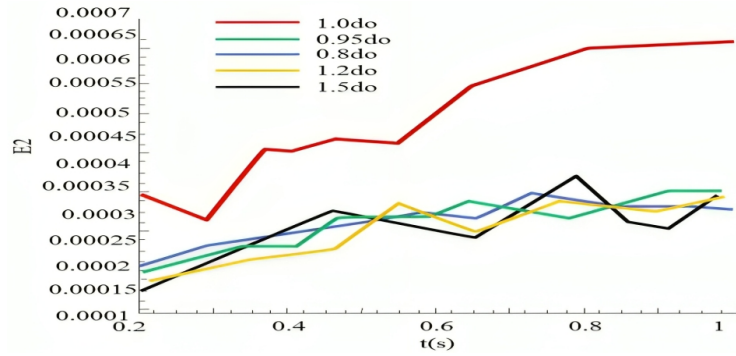


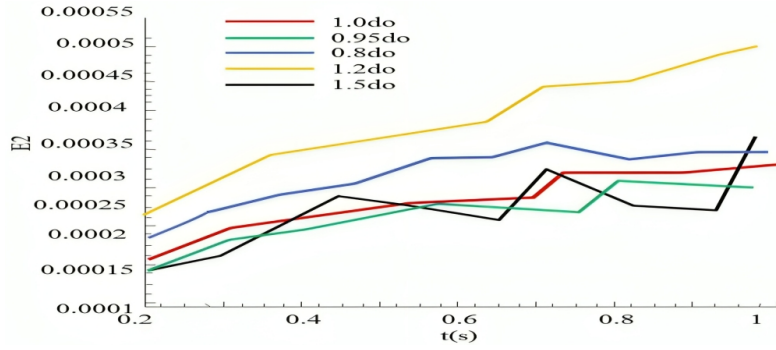
Figure 4. Comparison of absolute errors between the numerical solution and the true solution using three different kernel functions at the time of $t = 1$ under the condition of uniform distribution of news and public opinion information frames

In the numerical simulation, we take 801 news and public opinion information frames in the solution area, the corresponding space step is 0.05, and the time step is dt = 0.0001. The quintic spline smoothing function is selected, and the corresponding smoothing length $h = d_0$ is selected. Figure 6 a) shows the distribution of the absolute error between the numerical solution and the true solution obtained by applying the Strang split-corrected smoothed particle hydrodynamics (SS-CSPH) method to Equation (3) at different positions in the solution area at different times. It can be seen from this figure that the absolute error curve has a larger slope near the center of the solution area. In this case, the local encryption of the news and public opinion information frame can generally be used to reduce the slope of the error curve. Local encryption is performed in the range of $[0, 2]$ in this area, the space step size of the area without encryption is 0.05, and the space step size of the encrypted area is 0.025. Figure 6 b) shows the effect of local encryption through the comparison of 2-norm error curves. From this figure, it can be seen that local encryption has a better effect on reducing errors. Then, in the process of numerical solution using the Strang split-corrected smoothed particle hydrodynamics (SS-CSPH) method, if you encounter a place with a large slope of the error curve (indicating that the error in this area is relatively large), the local encryption method can be used to reduce the error in this area, so as to achieve the effect of reducing the overall error.

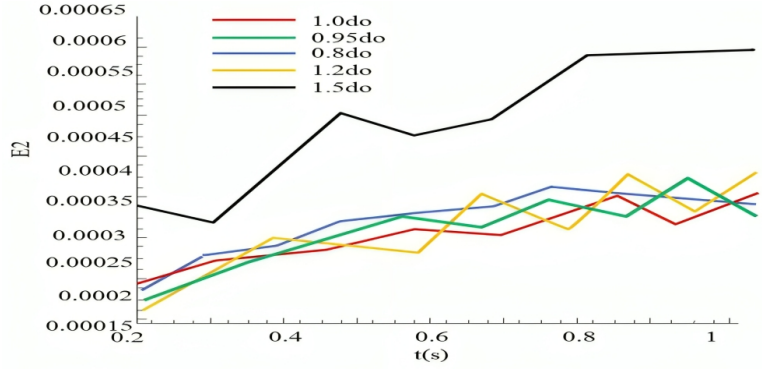
Equation (21) has no true solution, so the numerical results of Strang split-corrected smoothed particle hydrodynamics (SS-CSPH) are mainly compared with the results of the split finite difference method (SSFD) to verify the effectiveness of the Strang split-corrected smoothed particle hydrodynamics (SS-CSPH) method. The quintic spline smoothing function is selected, and the corresponding smoothing



a) Five-fold spline smooth function

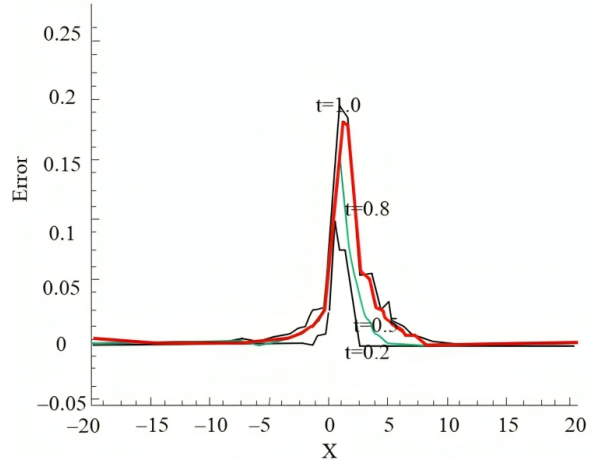


b) Three-fold spline smooth function

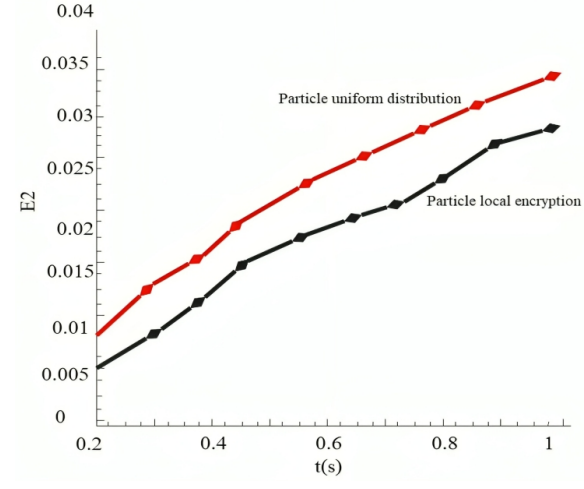


c) Gaussian-type smooth function

Figure 5. The comparison curve of the 2-norm error of the numerical solution obtained by the smooth function for different smooth lengths and the true solution in the case of uniform distribution of news and public opinion information frames in the numerical simulation



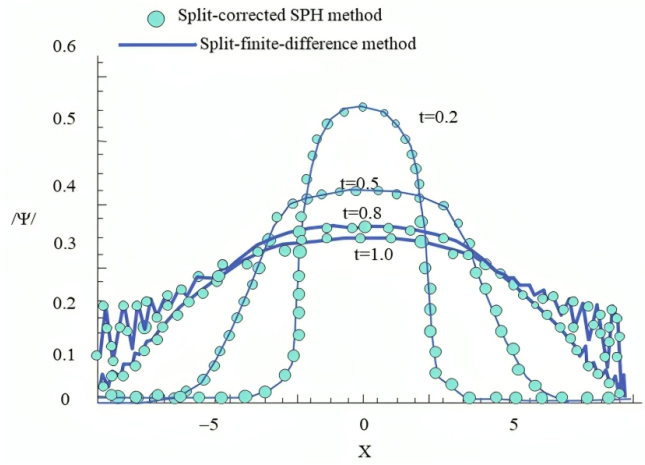
a) The distribution curve of the absolute error between the numerical solution and the true solution at different times



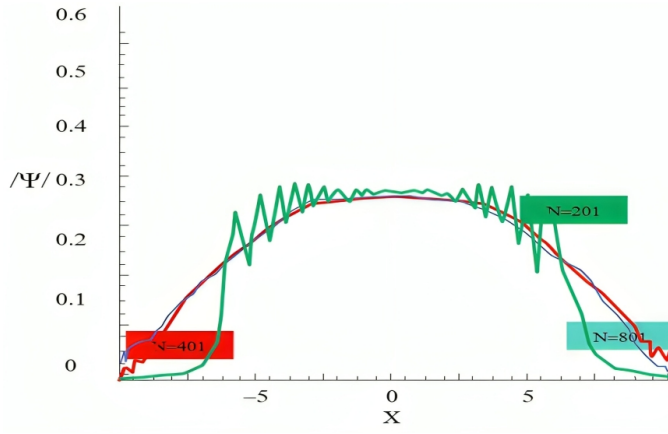
b) Comparison diagram of the 2-norm error of the numerical solution to the true solution before and after local encryption over time

Figure 6. Absolute error between the numerical solution and the true solution

length $h = d_0$ is selected. It can be known from Example 3 that local encryption can reduce the error of the numerical solution to the real solution. Through the Example 4, the influence of the gradual global encryption of news and public opinion information frames on the numerical accuracy of the Strang split-corrected smoothed particle hydrodynamics (SS-CSPH) method will be studied, that is, the numerical convergence of the Strang split-corrected smoothed particle hydrodynamics (SS-CSPH) method will be studied.



a) Comparison curves of the numerical solutions of the two numerical methods at different times



b) Numerical solution curve of SS-CSPH method under different frame numbers of news and public opinion information at time $t = 0.8$

Figure 7. Comparison curves of numerical solutions at different times

It can be seen from Figure 7 a) that the numerical solution obtained by Strang split-corrected smoothed particle hydrodynamics (SS-CSPH) is basically consistent with the numerical solution obtained by SSFD, and the corresponding spatial steps of the SS-CSPH method and the SSFD method are both 0.02. However, the time step corresponding to the SS-CSPH method is 0.0001, and the time step corresponding to the SSFD method is 0.001. According to the distribution of the numerical solution curves of different news and public opinion information frames (different spatial steps) in Figure 7 b), it can be known that the numerical solution of SS-CSPH gradually approaches a fixed numerical solution curve with the global gradual encryption of the spatial news public opinion information frame. That is, the numerical solution obtained by the SS-CSPH method satisfies numerical convergence and numerical stability. Schrödinger equation of $\alpha(t) = \frac{t}{2}$, $\beta(t) = -\frac{t}{2}$, $V(x,t) = 0$ (two-dimensional coupled variable coefficient nonlinear Schrödinger equation), namely:

$$\begin{cases} i\psi_t + \frac{t}{2}\left[\psi_{xx} + \psi_{yy}\right] - \frac{t}{2}\left(|\psi|^2 + |\phi|^2\right)\psi = 0, \\ i\phi_t + \frac{t}{2}\left[\phi_{xx} + \phi_{yy}\right] - \frac{t}{2}\left(|\psi|^2 + |\phi|^2\right)\phi = 0. \end{cases} \tag{21}$$

The true solution of this equation is:

$$\begin{cases} \psi = \dfrac{(1+i)e^{-t^2}\left(1+e^{x+\sqrt{5}y+\frac{\sqrt{3}}{2}t^2+\frac{2\pi}{3}i}\right)}{1+e^{x+\sqrt{5}y+\frac{\sqrt{3}}{2}t^2}}, \\ \phi = \dfrac{(1+i)e^{-it^2}\left(1+e^{x+\sqrt{5}y+\frac{\sqrt{3}}{3}t^2+\frac{2\pi}{3}i}\right)}{1+e^{x+\sqrt{5}y+\frac{\sqrt{3}}{2}t^2}}. \end{cases}$$

In the simulation of Example 5, the number of news and public opinion information frames is 201, the corresponding space step is 0.05, and the time step $dt = 0.0001$. The quintic spline smoothing function is selected, and the corresponding smoothing length $h = 0.9d_0$ is selected.

The numerical solution obtained by the SS-CSPH method is also in good agreement with the real solution when the distribution of two-dimensional news public opinion information frames is uneven. Moreover, the same error accuracy as the uniform distribution of news and public opinion information frames can be maintained in the entire solution area. That is to say, the SS-CSPH method has better applicability to the uneven distribution of news and public opinion information frames in high-dimensional problems.

Example 4. Equation (1) takes $\alpha(t) = \frac{1}{2}, \beta(t) = -10, V(\mathrm{x},t) = -\frac{1}{2}(x^2+y^2)$, namely:

$$i\psi_t + \frac{1}{2}\left(\psi_{xx} + \psi_{yy}\right) - 10|\psi|^2\psi - \frac{1}{2}\left(x^2+y^2\right)\psi = 0.$$

The initial condition is:

$$\psi^0(x,y) = \frac{2}{\sqrt{\pi}}(x+iy)e^{-\left(x^2+y^2\right)}.$$

This equation has no true solution, so the numerical results of SS-CSPH are mainly compared with the results of split finite difference method (SSFD) to verify the effectiveness of the SS-CSPH method. In this numerical simulation, for the SS-CSPH method, the number of news and public opinion information frames is 321, the corresponding space step is 0.05, the time step is dt = 0.0001, the quintic spline smooth function is selected, and the corresponding smoothing length is $h = 0.9d_0$. For the SSFD method, the space step is 0.08 and the time step is $dt = 0.001$.

3 PUBLIC OPINION MONITORING AND GUIDANCE SYSTEM IN THE PROCESS OF NEWS DISSEMINATION FROM THE PERSPECTIVE OF BIG DATA

When a user searches using a meta-search engine, the meta-search engine forwards the search request to its independent member search engine, and each independent member returns the result to the meta-search engine after executing the request. The meta-search engine performs secondary processing on these results and then feeds back the processed results to actual users. The workflow of the meta search engine is shown in Figure 8 a). Web crawler (Crawler) takes a web page as the center point and automatically crawls other pages according to the traversal method of graph theory. Web page information collection is based on this. Before the system starts web crawling, we need to select the initial URL and put the selected URL into a list, which is the initial URL list. The architecture of the web crawler is shown in Figure 8 b).

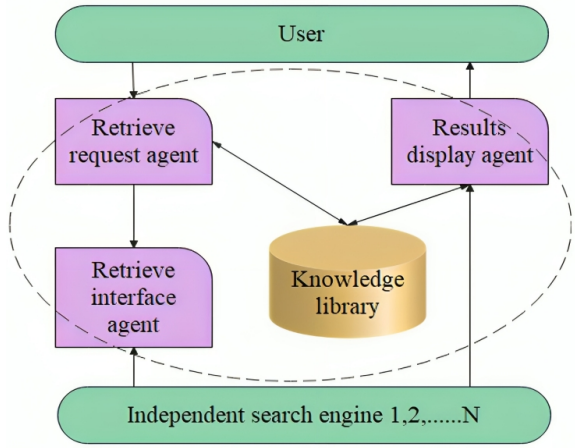
This paper proposes a template-based web page information extraction method. This information extraction method does not require repeated processing of the same type of web pages, so the efficiency is relatively high. In the network public opinion monitoring system, topic information needs to be extracted from a large number of web pages. Based on this, a Web information extraction method based on automatically generated templates is proposed. The schematic diagram of the template generation process is shown in Figure 9.

A total of 32 groups of simulation evaluations are carried out to evaluate the effect of the public opinion detection and guidance method proposed in this paper in the process of news dissemination, as shown in Table 1.

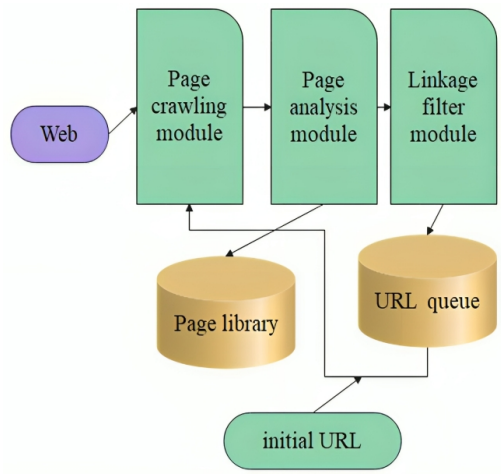
From the simulation evaluation results, it can be seen that the method of public opinion detection and guidance in the process of news dissemination proposed in this paper can play a certain role in the monitoring and guidance of news public opinion.

4 CONCLUSION

In the era of new media, advanced digital technology has accelerated the speed of media dissemination, allowing information dissemination to break the constraints of



a) Workflow of the meta search engine



b) Architecture of the Web crawler

Figure 8. Meta search engine and Web crawler

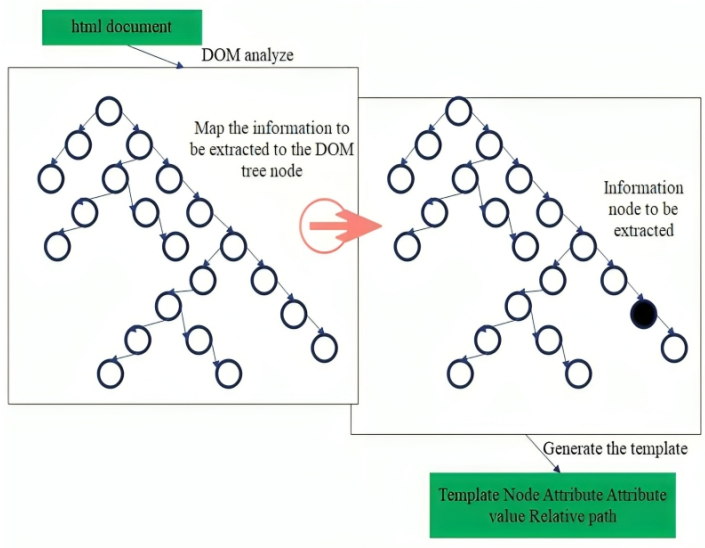


Figure 9. Schematic diagram of template generation process

Num	Public Opinion Monitoring	Public Opinion Guide	Num	Public Opinion Monitoring	Public Opinion Guide
1	84.593	76.799	17	83.814	79.548
2	85.534	81.320	18	87.076	75.558
3	82.830	74.874	19	84.544	77.341
4	88.041	79.953	20	86.772	74.316
5	83.154	81.297	21	82.795	81.639
6	88.831	81.734	22	88.404	81.178
7	86.756	78.126	23	80.816	80.762
8	83.587	81.085	24	80.544	78.115
9	86.801	77.635	25	82.318	73.808
10	83.251	79.402	26	81.177	79.818
11	83.269	79.458	27	87.465	80.306
12	81.369	78.709	28	86.825	74.306
13	81.569	79.319	29	80.381	73.883
14	86.455	76.547	30	86.903	81.088
15	85.731	80.137	31	88.823	74.347
16	84.329	73.839	32	86.434	78.259

Table 1. Simulation evaluation of the effect of public opinion detection and guidance methods in the process of news dissemination

space and time. At the same time, it also disrupts the roles of information publishers, disseminators and receivers, so that they are no longer strictly differentiated. The form of communication of new media is no longer just one-way communication. In the past, traditional media only served as information producers and disseminators. Today, every audience has the potential to be a publisher and communicator, not just a recipient. However, some journalists realize that copying each other, plagiarism is more efficient, and this vicious circle will only continue. Under the dual pressure of external temptation and internal squeeze, the mentality of practitioners will also change, and down-to-earth news production methods often do not get good results. From the perspective of big data, this paper conducts research on the detection and analysis of public opinion in the process of news communication, and analyzes the guidance of public opinion in news communication. From the simulation evaluation results, it can be seen that the method of public opinion detection and guidance in the process of news dissemination proposed in this paper can play a certain role in the monitoring and guidance of news public opinion.

REFERENCES

[1] Cui, L.—Xie, G.—Yu, S.—Zhai, X.—Gao, L.: An Inherent Property-Based Rumor Dissemination Model in Online Social Networks. IEEE Networking Letters, Vol. 2, 2020, No. 1, pp. 43–46, doi: 10.1109/LNET.2019.2952567.

[2] Shan, S.—Lin, X.: Research on Emergency Dissemination Models for Social Media Based on Information Entropy. Enterprise Information Systems, Vol. 12, 2018, No. 7, pp. 888–909, doi: 10.1080/17517575.2017.1293300.

[3] Kaur, S.—Kumar, P.—Kumaraguru, P.: Automating Fake News Detection System Using Multi-Level Voting Model. Soft Computing, Vol. 24, 2020, No. 12, pp. 9049–9069, doi: 10.1007/s00500-019-04436-y.

[4] Marwick, A. E.: Why Do People Share Fake News? A Sociotechnical Model of Media Effects. Georgetown Law Technology Review, Vol. 2, 2018, No. 2, pp. 474–512.

[5] Xu, D.—Pan, J.—Wang, B.—Liu, M.—Kang, Q.: Information Dissemination Model of Microblogging with Internet Marketers. Journal of Information Processing Systems, Vol. 15, 2019, No. 4, pp. 853–864, doi: 10.3745/JIPS.04.0126.

[6] Kelly, M. P.—Martin, N.—Dillenburger, K.—Kelly, A. N.—Miller, M. M.: Spreading the News: History, Successes, Challenges and the Ethics of Effective Dissemination. Behavior Analysis in Practice, Vol. 12, 2019, No. 2, pp. 440–451, doi: 10.1007/s40617-018-0238-8.

[7] Van Bavel, J. J.—Harris, E. A.—Pärnamets, P.—Rathje, S.—Doell, K. C.—Tucker, J. A.: Political Psychology in the Digital (Mis)Information Age: A Model of News Belief and Sharing. Social Issues and Policy Review, Vol. 15, 2021, No. 1, pp. 84–113, doi: 10.1111/sipr.12077.

[8] Ghosh, S.—Shah, C.: Towards Automatic Fake News Classification. Proceedings of the Association for Information Science and Technology, Vol. 55, 2018, No. 1, pp. 805–807, doi: 10.1002/pra2.2018.14505501125.

[9] Shu, K.—Mahudeswaran, D.—Wang, S.—Lee, D.—Liu, H.: FakeNewsNet: A Data Repository with News Content, Social Context, and Spatiotemporal Information for Studying Fake News on Social Media. Big Data, Vol. 8, 2020, No. 3, pp. 171–188, doi: 10.1089/big.2020.0062.

[10] Jadhav, S. S.—Thepade, S. D.: Fake News Identification and Classification Using DSSM and Improved Recurrent Neural Network Classifier. Applied Artificial Intelligence, Vol. 33, 2019, No. 12, pp. 1058–1068, doi: 10.1080/08839514.2019.1661579.

[11] Huang, H.—Wang, T.—Hu, M.—Dong, M.—Lai, L.: Node Attitude Aware Information Dissemination Model Based on Evolutionary Game in Social Networks. Mobile Networks and Applications, Vol. 26, 2021, No. 1, pp. 114–129, doi: 10.1007/s11036-020-01685-2.

[12] Shrivastava, G.—Kumar, P.—Ojha, R. P.—Srivastava, P. K.—Mohan, S.—Srivastava, G.: Defensive Modeling of Fake News Through Online Social Networks. IEEE Transactions on Computational Social Systems, Vol. 7, 2020, No. 5, pp. 1159–1167, doi: 10.1109/TCSS.2020.3014135.

[13] Borges-Tiago, T.—Tiago, F.—Silva, O.—Guaita Martínez, J. M.—Botella-Carrubi, D.: Online Users' Attitudes Toward Fake News: Implications for Brand Management. Psychology and Marketing, Vol. 37, 2020, No. 9, pp. 1171–1184, doi: 10.1002/mar.21349.

[14] Xiao, X.—Borah, P.—Su, Y.: The Dangers of Blind Trust: Examining the Interplay Among Social Media News Use, Misinformation Identification, and News Trust on Conspiracy Beliefs. Public Understanding of Science, Vol. 30, 2021, No. 8, pp. 977–992, doi: 10.1177/0963662521998025.

Wei Shi obtained his Doctorate in the Department of Communication. Currently, he is working as Lecturer. He received his graduation from the Shanghai University in the year of 2013. He also worked in the Nanchang Hangkong University. His research interests include history of communication and movie. He had published more than 10 papers in many reputed journals.

Weiwei Qin received her Master of Arts. Currently, she is a Postgraduate student. She is pursuing her Ph.D. in the Nanchang Hangkong University.

Computing and Informatics, Vol. 42, 2023, 1136–1157, doi: 10.31577/cai_2023_5_1136

CLUSTERING MINING ALGORITHM OF INTERNET OF THINGS DATABASE BASED ON PYTHON LANGUAGE

Fang Wan*

Business College, Nanchang Jiaotong Institute
Nanchang 330100, China
e-mail: ppap0112@163.com

Ying Liu

School of Artificial Intelligence, Nanchang Jiaotong Institute
Nanchang 330100, China
e-mail: liuying10100160@163.com

Abstract. In order to solve the problems of reading delay in data mining of the Internet of Things database, a clustering mining algorithm of the Internet of Things database based on Python language is proposed. We designed an improved crawler algorithm based on the open-source structure of scratch through Python language, judge the similarity of recruitment data topics in the Internet of Things database through Bayesian classifier, and crawl the recruitment data in the Internet of Things database: the number of keywords in the text space, the degree of keyword extraction, and the number of keyword data in the text space. The time series model is used to eliminate the delay of text features. On this basis, the semi-supervised learning and semi-cluster analysis method is used to construct the corresponding classifier, complete the adaptive classification process of the text data stream and realize the clustering mining of the Internet of Things database based on Python language. The experimental results show that this method has a low reading delay, and can mine the attention, number of posts and click time frequency of the Internet of Things database from which the recruitment data are obtained.

Keywords: Python language, Internet of Things database, data clustering, data mining

* Corresponding author

1 INTRODUCTION

The technical field of the Internet of Things is developing very fast, and there are many places to use [1]. There is more and more research on its related intelligent devices, and the type and quantity of products are also being used. Therefore, the application of this system or device has been quite common in people's daily lives. The rapid development of the Internet has driven the progress of big data. We can find a kind of useful data in a large amount of data. This mining method has brought great help to people and has become one of the aspects that many people are willing to explore and develop. Among them, Python is a popular program language in this field, because it has a lot of content, powerful technology and powerful computing power [2].

In reference [3], aiming at the problem of low clustering efficiency of moving objects in convergent pattern mining of the Internet of Things, a spatiotemporal feature mining algorithm based on pattern growth and multi-minimum support is proposed. Mining frequent and asynchronous periodic spatiotemporal motion patterns, modeling the position sequence and adding the time information to the model. Then the sequential patterns of asynchronous cycles are deeply and recursively mined to realize data clustering mining. Clustering methods can be applied to a wide variety of database systems. The information can take the form of quantitative, category, or interval-based information. Reference [4] proposed a two-level distributed clustering routing algorithm based on unequal clusters. The main idea is to reduce the data transmission distance between member nodes and cluster heads and alleviate the hot spot problem by distributing two cluster heads in each cluster, so as to achieve energy saving and load balancing. The sensor nodes are clustered by the clustering method. A further statistic frequently utilized to assess the effectiveness of clustering techniques is cooperative knowledge. It measures how distinct two descriptions for identical information are. The clustered sensor nodes can transmit data to the receiver through different paths with minimum energy consumption. Reference [5] proposed the first memetic algorithm to solve the balanced classical least square sum clustering problem. The algorithm combines responsive threshold search and a backbone-based crossover operator to generate offspring and realizes data clustering mining. However, in the process of practical application, the above methods are prone to problems such as long response time due to the influence of data type and data scale in the Internet of Things database.

Python can develop languages for different objects. It is very powerful in computing and has a wide range of code resources. Therefore, Python has gradually become a tool used in the direction of data mining, and more and more tools are used for development [6]. Using Python to conduct this kind of work is not only convenient and requires less, but also involves popular relevant tool libraries, which will greatly reduce the steps of each link in the work, which will save a lot of time for researchers to focus on the design and development of data mining direction, so as to get better and more accurate results [7]. The foundation of this method is machine learning. It arranges the parameters or elements in an information source into

predetermined types or groups. Methods for data mining include synthetic neural networks, selection orchards, analytics, linear regression, and more. Therefore, this paper studies the clustering mining algorithm of the Internet of Things database based on Python language.

2 CONSTRUCTION OF CLUSTERING MINING ALGORITHM FOR INTERNET OF THINGS DATABASE BASED ON PYTHON LANGUAGE

Among the technologies for processing big data in the existing Internet platforms, many have not been optimized enough in terms of technical activity, and the reflection effect from users is not good. At the same time, there is no such technology that can meet the whole platform operated by users in the market. The current service platforms have relatively high prices in the market and lack the framework of technology types suitable for small companies or scientific research. Contract administration services, e-signature techniques, online marketing, consumer connection administration (CRM), payroll systems, interaction as well as collaborative techniques, as well as access control techniques, are just a few examples. In the past, private organizations relied on newspaper advertisements or posters at the end of the road. They now have access to considerably more potent tools for expanding their firm. Python has now become a well-developed and most frequently used program in the database field [8, 9]. Because it has very rich resources and powerful computing functions, it can help to complete the work more effectively in the mining process. To access greater reserves, underwater miners are more costly. Shallower as well as less valued resources are often mined using subsurface methods. Therefore, this paper uses Python language to mine, analyze and model the recruitment data in the Internet of Things database. The specific framework is shown in Figure 1.

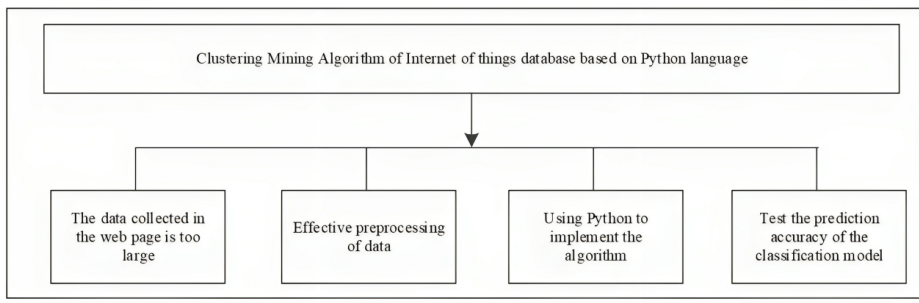


Figure 1. Algorithm framework

2.1 Improved Python-Based Scratch Crawler

In the traditional mode, the general open source crawler architecture of Sweep can not crawl the exclusive content. Therefore, the open source crawler structure based on Sweep is constructed, and the Bayesian classifier [10] is selected to judge the topic similarity. The clustering technique uses an automatic method in which the data is not labelled and also uses the solution to a challenge determined by the algorithm's expertise from a set of practice issues. Information grouping from all network members is the responsibility of every member node. Whenever a frame of information from every membership is obtained, the CH applies data acquisition before sending the frames towards the access point [11]. Through the search method, grab the Internet of Things database data in the Internet of Things database, analyze the recruitment data in the Internet of Things database through the clustering algorithm, maintain the node weights of the Internet of Things database at the present stage according to the analyzed contents, analyze the link weights of the child URLs according to the recruitment topic relevance module, and eliminate the topics less than the URLs node weights with the filter unit after calculation. Deduplicate the URL in the scratch structure, and determine the subject big data related to the Internet of Things database data through the following Bayesian classifier.

1. Set the set as $X = \{x_1, x_2, \ldots, x_m\}$, where the attribute in x is $a_1 - a_m$.
2. Set $Y = \{y_1, y_2, \ldots, y\}$ as the set of existing categories.
3. Calculate the probability $P(y_1 \mid x), P(y_2 \mid x), \ldots, P(y_n \mid x)$ in turn.
4. Calculate the maximum probability $P(y_k \mid x)$, the highest probability category has X and $x \in y_k$.

Through Bayesian theory, when $P(X \mid Y)$ is calculated, $P(Y \mid X)$ can be obtained. Therefore, Y is approximately equal to A and X is approximately equal to B. When B is divided into different attributes, each type of attribute is calculated in the following form:

1. Get a training sample set.
2. We can know the category of the sample set, and it is the set of items to be classified.
3. Through statistical analysis in the form of formula (1), the probability estimation value $P(x_m \mid y_n)$ of classification conditions is obtained.
4. If there are all characteristic attributes with independent conditions, it can be calculated by formula (1):

$$P(y_i \mid x) = \frac{P(y_k \mid x) P(y_i)}{P(x)}. \tag{1}$$

Since the denominator $P(x)$ is usually set as a constant, the maximized molecule can be calculated. At the same time, since each feature has conditionally independent attributes, therefore:

$$P'(y_i \mid x) = P(y_i) \prod_{k=1}^{n} P(y_k \mid x). \tag{2}$$

Through the above construction process of Bayesian classifier, we can judge the similarity of recruitment data topics in the Internet of Things database. When there are type feature attributes with larger weight, the higher the similarity of recruitment data topics in the Internet of Things database.

2.2 Data Preprocessing of Internet of Things Database

In this design, Python language will be used as the processing basis [12, 13], and this technology will be used to optimize the original data stream classification method. Individual categorization as well as ensemble categorization are the two techniques used to classify big quantities of information. The one-classification approach is quick and uses minimal storage during processing. At the same time, select the appropriate technology to complete the data flow processing, convert the original relational database into non-relational database, and improve the use effect of classification methods. Information is kept in a non-relational system, sometimes referred to as a NoSQL dataset. There are no columns, lines, cardinality, or database objects, except for database systems. Rather, models are appropriately tailored to the unique needs of the sort of data being saved used by the non-relational databases.

2.2.1 Text Data Stream Data Acquisition

The recruitment data in the Internet of Things database is complex and changeable, and the types vary greatly. When selecting target data, careful filtering and screening is required [14]. In this study, the data acquisition process is controlled by a formula, and the specific contents are as follows.

Set the unmarked data stream collected in the Internet of Things database, E contains N data records, and (x_w, y_w^{α}), where x_i represents a data label of w-dimensional attribute in the data, y_w^{α} represents the category label of w-dimensional attribute in the data, and α represents unknown items in the category. In order to build the recruitment data collection process in the Internet of Things database into the form of model, the data collection results are expressed as the set of marked data block F_i^g and unmarked data block F_i^h, and expressed by $E = \{F_1^1, F_2^g, \ldots, F_{m-1}^h, \ldots, F_m^n\}$.

Divide the collected i data blocks into positive data blocks and negative data blocks respectively. On the premise of ensuring the comprehensiveness of the data, set the first f data $f_1^1, f_2^g, \ldots, f_{m-1}^h, \ldots, f_m^n$ in the collected data as marked data

blocks [15, 16], and the remaining $f_{1+n}^{1}, f_{2+n}^{g}, \ldots, f_{m-n}^{h}, \ldots, f_{m}^{n}$ as unmarked data blocks. The types of data blocks can be divided by using formula (3):

$$y^{z} = P'\left(y_i \mid x\right) \arg\min\left\{\operatorname{dis}\left(x, y^{\varepsilon}\right), \operatorname{dis}\left(x, y^{n}\right)\right\}, \tag{3}$$

where $\operatorname{dis}\left(x, y^{2}\right)$ represents the distance between the unlabeled data block and the center point of the positive data block.

$$\operatorname{dis}\left(x, y^{8}\right) = \left\|x, y_1^{8}\right\|_2, \tag{4}$$

where, y_1^8 represents the center point of the positive data block. Similarly, the distance $\operatorname{dis}\left(x, y^{n}\right)$ between the unlabeled data block and the center point of the negative data block can be obtained.

According to the above process, the recruitment text data stream data in the Internet of Things database is obtained and used as the data basis in the subsequent classification process.

2.2.2 Data Stream Preprocessing and Mining

In order to better process the recruitment text data in the Internet of Things database, Python language and the Internet of Things database crawler technology [17] are applied to the process of data mining. Crawler search results include, for instance, Google and Yahoo. Crawlers have the benefit of containing a large quantity of documents. Search the keywords through the Internet of Things database crawler technology, get the corresponding data, integrate it into the form of database, and carry out preprocessing and mining. A web crawler, also referred as a retrieval system or internet spiders, is a software program utilized to systematically explore as well as analyze the contents of websites as well as similar online data. Because the Python language used in this design has a unique use environment, the original SQL database is transformed into PyMongo library, which is a non-relational database. In the Connections Graph, either right-click on a service, databases, or collecting and choose SQL Conversion from the context menu, or choose the SQL Conversion arrow in the taskbar. The suggested method for interacting with MongoDB from Python is PyMongo, a Python package that includes capabilities for doing so. Although MongoDB maintains information in BSON language both locally and remotely, users can still conceive of MongoDB as a JSON library. Different from the databases currently in use, PyMongo library can store text data from various sources and in various forms. Generally speaking, PyMongo offers a comprehensive selection of services that you are able to use for interaction with a MongoDB host. It offers the ability to access, extract data, update as well and remove information, in addition to executing network operations. At the same time, it improves the flexibility of the data model, makes the reading and writing of data easier, and facilitates the secondary expansion of data.

In this study, the text data is transformed into machine recognizable text and stored in vector space to complete the data stream preprocessing process. In order

to obtain the text features of the Internet of Things database data, the keywords in the data are extracted in the form of extraction according to the number and criticality of keywords [18, 19]. Data processing is the technological method of modifying information from another format, standard, or architecture to a different one without altering the information's contents. This is often done to make the information more usable by users or apps or to enhance the accuracy of the information. Access the URL string injected by SQL in the Internet of Things database, describe the string through a single text and divide the data in the form of word segmentation. After division, the space vector can be designed, which corresponds to the URL string. In this space, each feature can be described through the characteristic keywords added by SQL. The characteristics of SQL are simple to acquire. The following features make SQL a particularly useful and approachable vocabulary: wide range of instructions, database objects, movable vocabulary, reunites, unification, considered ideal for the client relationship, and interoperability. Among them, the inverse document rate can be determined by weight and analyzed according to word frequency [20].

1. **Word frequency:** Because the URL string is usually output in the form of ASCII code, and there is no rule in the string, when starting to segment the URL access data, it is necessary to segment the word in the form of "%", "&" and space, so that the data after the word segmentation can be left.

   If the keyword appears in each spatial vector feature, the number of times the keyword is found. The number of times after sorting is described by the word frequency. Keyword research is the method of locating these phrases. You will be capable of carrying out relevant keywords alone following completing this course. If there is no keyword, the word frequency is zero. The greater the number of keywords in the sample, the greater the importance. The process of adjusting the word frequency to the spatial vector model is represented by formulas (5) and (6):

$$T_1 = \frac{\text{tf}}{doc\_\text{length}}, \tag{5}$$

$$T_2 = \frac{tf}{\max ff}, \tag{6}$$

   where T_1 and T_2 represent features 1 and 2; tf indicates word frequency; doc_length indicates the length of the statement; max ff indicates the maximum word frequency in the statement. The process of transferring word frequency to spatial vector model can be carried out through formula (5) and formula (6).

2. **Reverse document rate:** Usually, the spatial vector is composed of word frequency and inverse document rate. Through the w-dimensional space, it can describe the spatial vector of each access data. The number of SQL injection keyword categories in the URL access data can determine the proportion S of

statements covering keywords in all statements. According to the above calculation of word frequency, the contribution degree of each word in the sentence can be obtained. When uploaded to the spatial vector, the interval degree between two sentences can be obtained [21]. When the number of keywords in a large number of data samples gradually decreases, it means that the data samples of the keyword are gradually reduced. Therefore, the higher the inverse document rate of the keyword, indicating that the word can distinguish sentences to a greater extent. The inverse document rate is described by I, therefore, I can be calculated by formula (7):

$$I = \ln\left(\frac{S}{df}\right) + 1. \tag{7}$$

If you want to realize that the reverse document rate is within the $[0, 1]$ range, you can calculate it through formula (8):

$$I_{new} = \frac{\ln\left(\frac{S+0.5}{df}\right)}{\ln S}. \tag{8}$$

Through formula (8), the calculation of inverse document rate can be completed, and its value represents the feature vector of a keyword.

Since the data in the form of a local domain name is saved, the data in the form of HTML source code will be cleaned for the data in the form of local domain name. According to the text space vector model constructed in this paper, the word frequency features of Internet of Things database data are extracted, and the final form is stored in HDFS (Hadoop Distributed File System) distributed database. An algebraic paradigm called the vector space paradigm treats items (including words) as coordinates. This enables determining word resemblance or the relevancy of a user's query or content simply.

In the process of data flow cleaning, deal with the abnormal items in the data collection, correct the data whose attribute value is missing and is not related to the data flow due to other reasons and correct the errors in the data collection. Because the research object is the Internet of Things database data, its types are more complex. Therefore, outliers in the data stream need to be processed to ensure the accuracy of data classification. In data cleaning, the data with large differences in average value in the data stream shall be eliminated to avoid data pollution. The most common causes of poor data integrity are improper data entry typos, erroneous material, misleading data, and so on. Each organization needs to perform data analysis. At the same time, set the corresponding data rules to restrict the data preprocessing results and data mining results. For multidimensional data such as recruitment data in the Internet of Things database, data preprocessing will relatively reduce the data dimension, resulting in the inability to accurately define the data in the process of data analysis. When this data reaches a certain amount, the high latitude of the data

will increase the difficulty of data analysis and make the overall proportion of the data unbalanced. Organizations have the difficulty of expanding analysis process due to the continually growing storage capacity. As the information accumulates, it gets more and more complex to analyze and provide various documents. Therefore, in the process of data mining, adaptive classification method is selected to improve the accuracy of clustering mining of the Internet of Things database.

3 ADAPTIVE CLASSIFICATION OF TEXT DATA STREAM

The traditional supervised learning algorithm has poor data ability for data, which is easy to cause a huge waste of data resources. Using machine learning algorithm to complete the classification of data flow, the classifier has low generalization ability, which can effectively improve the effectiveness of data flow processing. The method of semi supervised learning and semi cluster analysis [22, 23] is used to construct the corresponding classifier. Semi-supervised aggregation techniques are those that can be used with partly labelled information or data with different kinds of outcomes metrics. Creating groupings or clustered whilst making sure that the data are as comparable as feasible in each category is the overall goal of hierarchical clustering in advertising. Semi supervised learning is a new method in the field of machine learning. For a given data preprocessing result get J, it can be expressed as:

$$J = (x_1, y_2), \ldots, \left(x_{|g|}, y_{|g|}\right), \left(x_{|g|+1}, y_{|g|+1}\right), \ldots, \left(x_{|g|+|h|}, y_{|g|+1}\right), \tag{9}$$

where, $|g|$ represents the number of labeled samples, $|h|$ represents the number of unmarked samples, and $|h| + |g| = n$. Semi supervised learning is to classify and process the data with missing identification in the sample. By service-learning, individuals acquire to think critically as well as consider their experiences. These abilities include the capacity to connect seemingly unrelated aspects of an encounter in significant manner, to look for trends as well as higher purpose in data, or to assess and evaluate situations. The data is shown in Figure 2.

According to the above formula, assuming that the marked data and ummarked data in the data are mixed distributed according to Gaussian distribution, then:

$$r(x \mid \delta) = \delta \sum_{i=1}^{n} \varepsilon_i \left(x_{|g|+|h|}, y_{|g|+1}\right), \tag{10}$$

where ε_i represents the mixing coefficient of the data stream and δ represents the distribution coefficient. The Bayes Theorem is the foundation of Bayesian categorization. The economic analyzers are Bayesian predictors. The likelihood that a specific packet corresponds to a specific class is one example of a class participation likelihood that can be predicted by Bayesian processors. According to Bayesian classification theory, if the recruitment data is class q in the data flow, the posterior

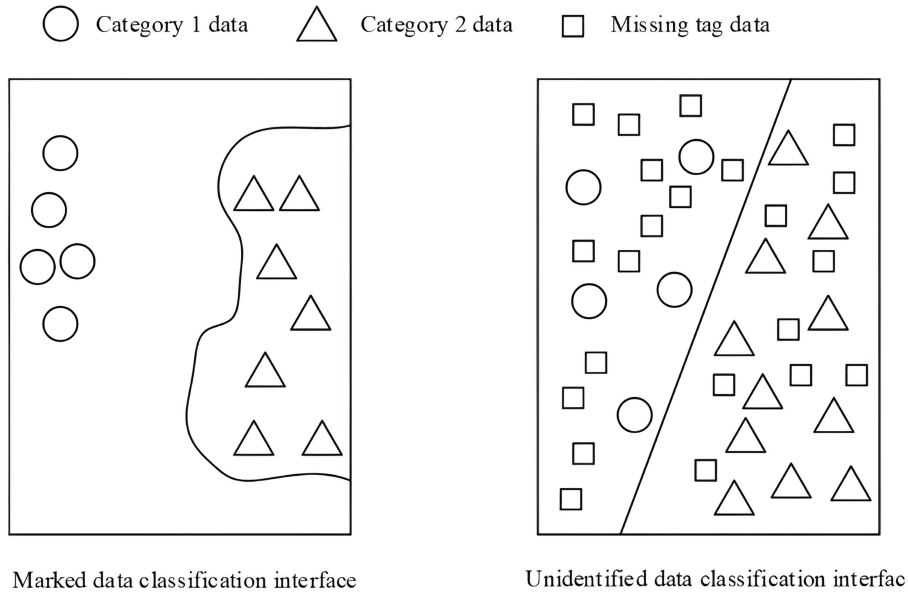


Figure 2. Data flow classification hyperplane

probability is the largest. The specific calculation formula is formula (11):

$$p = \frac{\varepsilon_i r(x \mid \delta)}{y^x \sum_{i=1}^{n} r(x \mid \delta)} \tag{11}$$

The unmarked data in the data stream is estimated through formula (11), and the corresponding ummarked data is obtained through the marked data, so as to improve the accuracy and ensure the use effect of the classifier.

A dynamical structure is found to suit a particular message or generalized linear of information is a time series description, often referred to as a signaling prototype. Multimodal algorithms can be created when the time series is multidimensional. In addition to the above process, according to the big data text features extracted by the text space vector model, a time series model is designed to arrange the data to remove the delay in the data. Let Z_i be the multidimensional random variable at this stage, and $Z = \{Z_1, Z_1, \ldots, Z_N\}$ represents the number of recruitment data in the Internet of Things database at this stage. When it is within the data value range of routing link layer, calculate the similarity characteristics of recruitment data through formula (12):

$$\rho = \varpi\left(x, y^g\right) + p \prod_{k=1}^{n} u\mathrm{P}\left(y_k \mid x\right), \tag{12}$$

where u and ϖ represent the response frequency and baud rate of the characteristic sequence exclusive to the recruitment data. According to formula (12), the time series can be constructed through time nodes, so as to eliminate the delay characteristic. Differentiating is arguably the easiest way to detrend a linear model. The quantity at the present rate phase is determined as the differential among the initial remark as well as the measurement at the prior data increment, namely, and a novel sequence is created.

The above contents are integrated and integrated with the adaptive classification method of data flow. So far, the design of clustering mining algorithm of the Internet of Things database based on Python language is completed.

4 EXPERIMENTAL ANALYSIS

4.1 Experimental Setup

This method is applied to a recruitment website to crawl the recruitment big data of the website, and the performance of this method is verified by experiments. At the same time, the data mining clustering method based on multiple minimum support of pattern growth in reference [3] and the data mining clustering method based on response threshold search in reference [5] are compared with this method. Minsup, or sequential pattern extraction, operates under the implied premise that each object has the identical quality (i.e., frequency). The regulations concerning rare Things will not be recognized if the minsup level is sufficient. Reaction criteria describe the probability of responding to stimulus related to a given task. People with low thresholds do activities with less stimuli than people with high thresholds.

In order to test the clustering mining effect of this method, the experimental test is carried out by using Windows 10 operating system. The specific experimental environment is as follows: Hardware environment: PC: CPU 30 GHZ, 8 GB memory and 500 GB hard disk; Development language: using Java language, it is relatively simple, flexible, portable and superior in performance.

Development tools: Eclipse is an extensible development platform with the advantages of easy operation. It is a development tool often used by Java.

Database development tool: MS SQL Server 2008, with high intelligence and high development efficiency. In the above experimental environment, the proposed algorithm is used to crawl the data from the recruitment website and take it as the data source of this experiment. The URLs on the website that a web search crawler may visit are specified in a robots.txt file. This is mostly intended to prevent the website from becoming overloaded with queries; it is not a method of taking a website off from Search. First, the crawler sends an access request and automatically saves its content if it meets the access requirements; Secondly, the crawler analysis module is used to obtain the remaining links in the crawled pages and take them as the subsequent crawling targets. Web crawlers are applications created to traverse the network, obtain information, organize it, then analyze it to speed up results. Crawlers are a clever answer to huge amounts of information and a driving

force behind significant developments in the cyber security sector. The web crawler framework is shown in Figure 3.

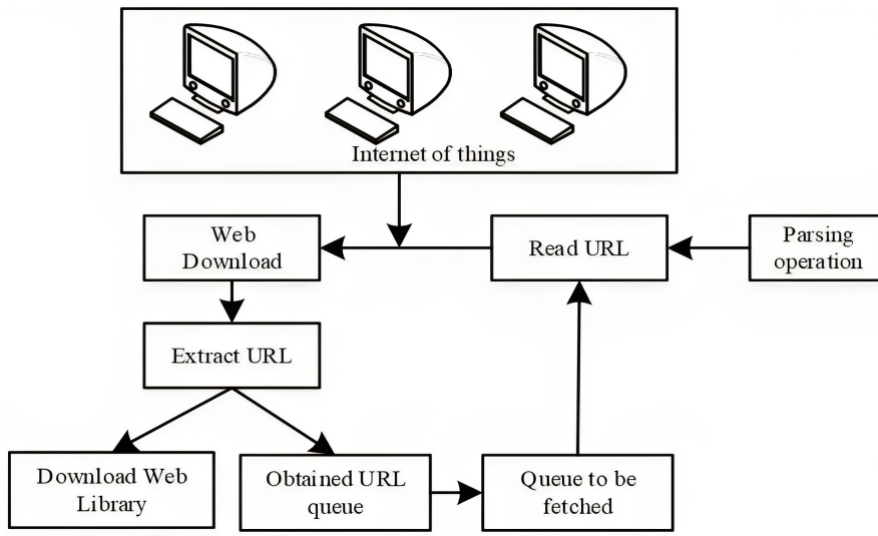


Figure 3. Schematic diagram of crawler frame

Similar to the working principle of users using browsers, the crawler crawling process also processes requests. Search engine crawlers use the technique of crawling to find provided by web resources. Whenever a person enters a search function, the appropriate outcomes are displayed on the web browser due to crawling, which is the process by which search engine crawlers scan the web sites or store a record of every data on indexed databases. Take the operation of rendering a web page by the browser as an example. When a user opens a recruitment web page, the browser will initiate a request, and the server will respond to the request. At the same time, the browser will display the parsing request on the page.

Develop crawler workflow through crawler framework. A web crawler finds URLs, then reads, analyses, and categorizes web sites. They discover connections to other websites anywhere along route as well as add those to the database of websites to crawl later. Select the recruitment web page with high value as the initial running target, and save the web page address and its corresponding path name to the downloader. For the recruitment webpage downloaded to the local, it is processed in three parts: first, save it to the preset page library and wait for index processing; Second, mark the page as a crawling page and add it to the crawling queue; Third, analyze the downloaded recruitment page and compare it with the capture queue. If there is no such page in the queue, transfer the recruitment page out of the queue and continue to wait for the scheduling instruction. According to the above process, the crawler runs automatically, completes all crawling work, and obtains

the recruitment data mining results. The whole process of clustering and mining recruitment data in the Internet of Things database using the above procedures is shown in Figure 4.

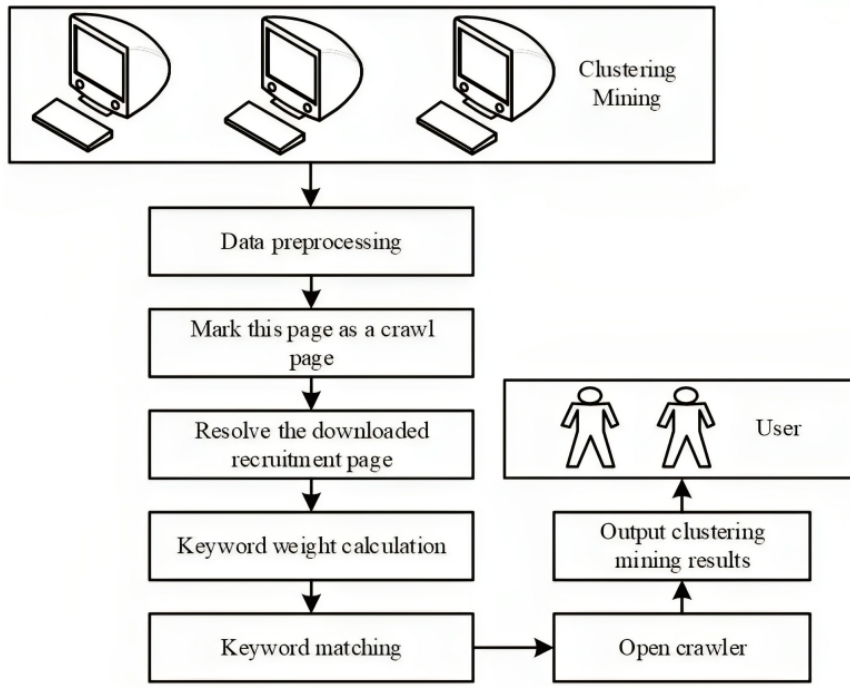


Figure 4. Schematic diagram of recruitment data clustering mining in the Internet of Things database

According to the process shown in Figure 4, obtain recruitment data and set that each data flow contains multiple and different numbers of sample data blocks. The specific data information is shown in Table 1.

	Test Data Set 1	Test Data Set 2	Test Data Set 3
Number of data blocks/piece	30	60	50
Number of attributes/piece	20	25	35
Number of concepts/piece	6	7	5
Number of time points/piece	45	25	30
Total data/piece	650	750	1 150
Important information/article	15	25	30

Table 1. Test data set setting

4.2 Result Analysis

Analyze the crawling ability of recruitment data of the Internet of Things database with different methods, and conduct experiments for various indicators. It all comes down to gathering and evaluating data utilizing modelling, computer vision, and statistical methods in order to provide the most accurate predictions about what might occur in certain situations. The experimental results are shown in Table 2.

Data Crawling Capacity	Paper Method	Reference [3] Method	Reference [5] Method
Climbing speed	About 300 pieces/min	About 270 pieces/min	About 260 pieces/min
Pertinence	Very high, able to crawl specific data	High, more data can be obtained	High, easy to lose some important data
Expansibility	Strong ability to crawl data associated with keywords	Strong, only the specified keywords can be crawled	Strong
Generality	Very strong	Strong	Strong for simple changes
Flexibility	Very strong	Strong	Strong, can only change with the changes of the web page
Crawling performance	It has strong crawling performance and can parse html files and xml files at the same time	When parsing html pages, you need to use other libraries	It mainly constructs the web scraper, and the crawling performance is poor
Crawling ability	Frame structure, which can crawl to the corresponding web page data and structured data, and the crawl speed is fast	It can extract data, but it cannot be used as a crawler independently	The speed of crawling the corresponding web page data is slow

Table 2. Recruitment data crawling capacity of the Internet of Things database with different methods

According to Table 2, through the analysis of data crawling ability of different methods, the data crawling speed of this method is fast, and about 300 data can be crawled per minute, while the crawling quantity of the other two methods is lower than that of this method in every minute, and the crawling performance of this method is higher than that of other methods. Therefore, this method can effectively realize the crawling of recruitment data in the Internet of Things database, provide a favorable basis for the subsequent mining of this kind of big data.

Basement, open-surface (pit), subsurface, as well as in-situ mining are the four basic types of extraction. Since it takes a certain time to mine, five feature data groups are selected to analyze the reading delay of different mining methods through the comparison of three methods. The experimental results are shown in Figure 5.

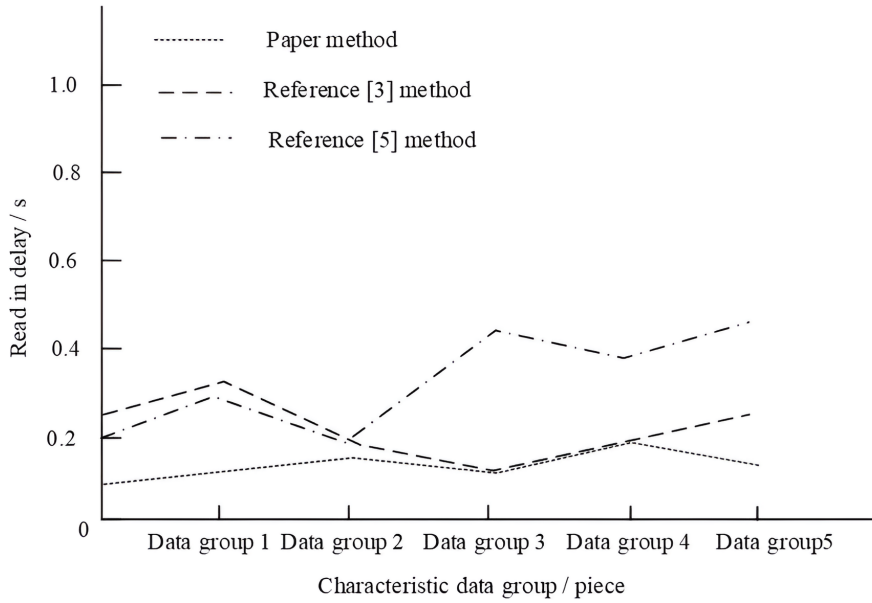


Figure 5. Comparison of read in delay of different methods

It can be seen from Figure 5 that due to the inconsistency of data in each group, there are different reading delays in each group. Among them, the reading delay of reference [5] method in data group 3, data group 4 and data group 5 remains the highest, up to 0.8 s, while the reading delay of reference [3] method in data group 1 and data group 2 is the highest, up to more than 0.4 s, while the method in this paper maintains the lowest reading delay in each group of data mining, And they are kept below 0.2 s. Therefore, this method can quickly realize big data mining.

An experiment is conducted on the data keywords of the Internet of Things database mined by using this method. By analyzing the actual attention of the keyword in different dates and the attention obtained by using this method, the big data mining ability of this paper is obtained. The analysis results are shown in Figure 6.

According to Figure 6, with the gradual increase of the date, the attention of the recruitment data gradually increases. When it reaches the 15^{th} day, the attention of the recruitment data gradually decreases until the recruitment data is no longer

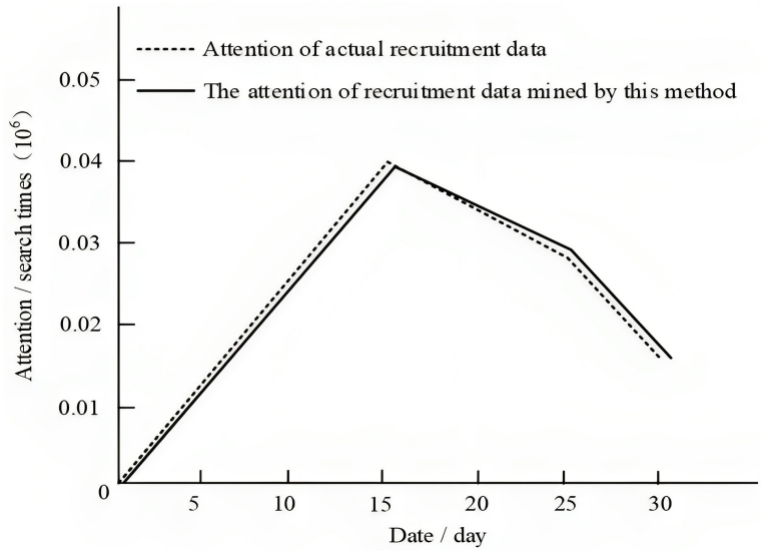


Figure 6. Analysis of recruitment data attention

concerned. The test shows that the attention of the recruitment data mined by this method is almost in line with the actual attention, and there is no large deviation. Therefore, this method can effectively mine the recruitment data in the Internet of Things database.

According to the number of posts generated by recruitment data keywords in the Internet of Things database in 24 hours, analyze the number obtained after mining with different methods. The analysis results are shown in Figure 7.

According to Figure 7, the number of new posts in a day changes from increase to decrease with time, and the number of new posts is the highest around 11:00 and 19:00. After mining by different methods, the change of the number of posts obtained by this method is completely similar to that of the actual posts, while the number of posts obtained by the method of reference [5] does not meet the standard of the actual number of posts, The number of posts obtained by the method of reference [3] is completely higher than the actual number. Therefore, the mining process of this method is more accurate and can fully mine the number of new posts associated with this keyword.

Analyze the click time frequency within 1 h after the recruitment keyword is generated, and compare it with the click situation obtained after mining by this method. The analysis results are shown in Figure 8.

According to Figure 8, the click time frequency of the keyword gradually increases within 1 h and decreases after 30 min. The mining results of this method are consistent with the actual click time frequency, which can clearly mine the click situation of the keyword.

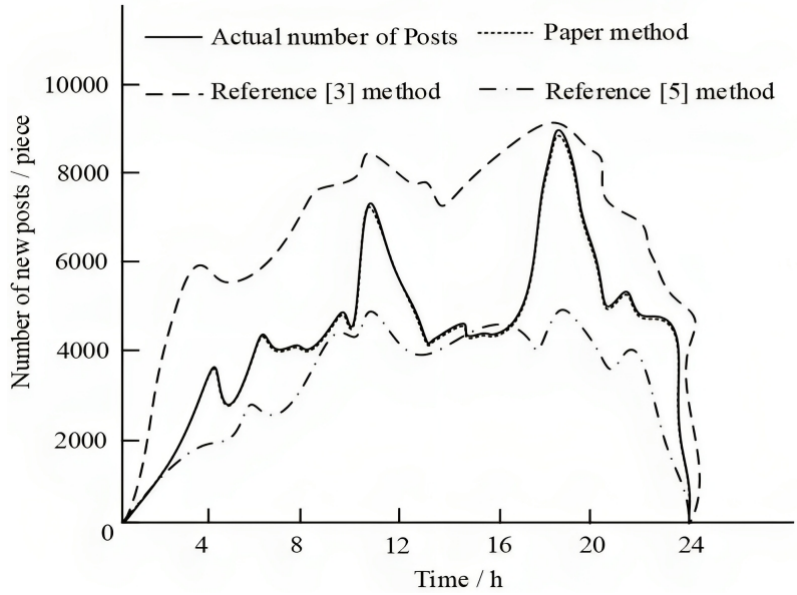


Figure 7. Number of new posts mined by different methods

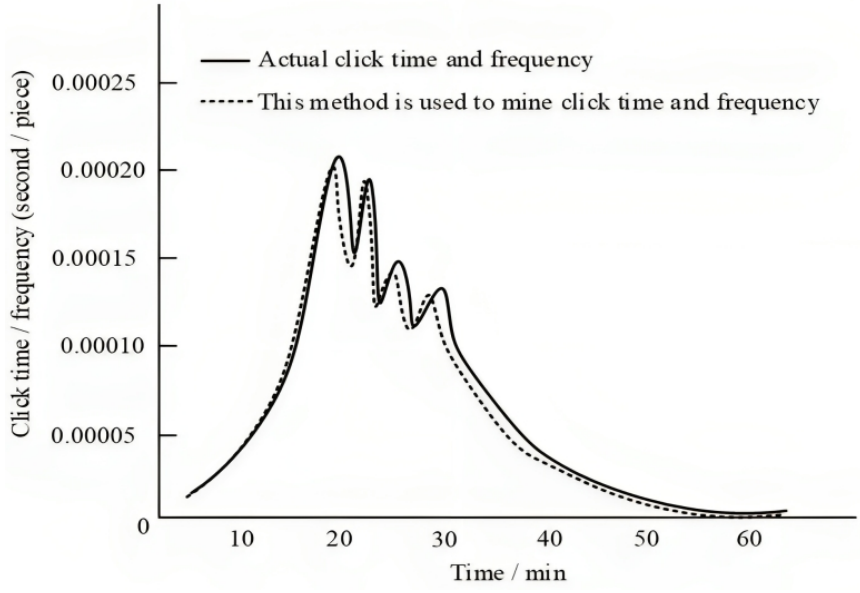


Figure 8. Comparison of keyword click time and frequency

Five retrieval experiments were conducted in the recruitment website to obtain the accuracy and recall of the three methods. Recall is the modeling measure we utilize to choose the optimal platform whenever there is a massive expense connected with False Negative, according to the identical concept. Several items and components can be structured using packaging, that contributes to a well-organized structure of information and makes subdirectories or sections simple to retrieve. The experimental results are shown in Table 3.

Number of Experiments	Paper Method		Reference [3] Method		Reference [5] Method	
	Accuracy [%]	Recall Rate [%]	Accuracy [%]	Recall Rate [%]	Accuracy [%]	Recall Rate [%]
1	90.6	63.1	83.2	60.1	82.7	61.4
2	94.8	68.2	87.3	63.9	80.3	66.3
3	93.1	72.8	78.8	62.2	82.9	69.9
4	92.2	72.7	79.6	60.2	89.2	65.3
5	95.9	78.8	80.9	65.9	81.5	68.1

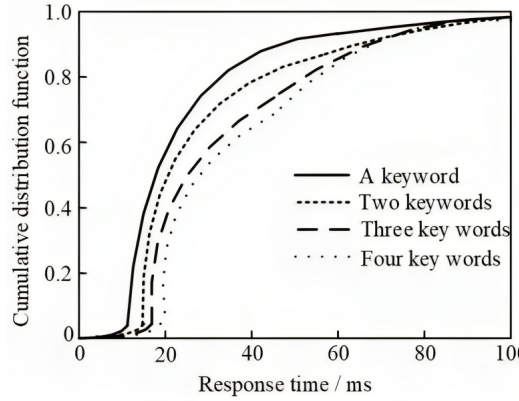
Table 3. Retrieval accuracy and recall of different methods

It can be seen from Table 3 that in the five retrieval tests, the performance of this method is more stable, and the accuracy and recall are significantly higher than the other two methods. This is because before the start of the crawler, all resource tags are preprocessed to remove low-frequency words and stop words, retain key words, and improve the retrieval accuracy to a certain extent. When the above experimental conditions remain unchanged, the response time of clustering mining of the three methods is compared, and the results are shown in Figure 9, respectively.

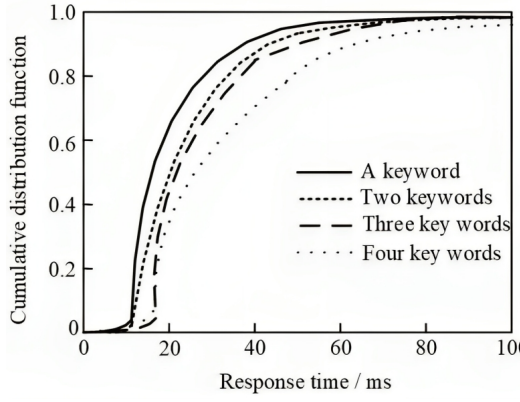
As can be seen from Figure 9 (9 a), 9 b) and 9 c)), the cumulative distribution function curves of the three methods show a smooth beginning, then rise significantly, and finally reach the mean value. However, when there are fewer keywords in the methods of reference [3] and reference [5], the required response time is shorter, but the clustering mining delay increases with the increase of keywords. The method in this paper is not sensitive to the number of keywords. It will get a response at about 40 ms, and there is almost no significant difference between the four curves. This is because the crawler route design is more reasonable, the crawler route with low weight is removed, and the response speed of clustering mining is accelerated.

5 CONCLUSION

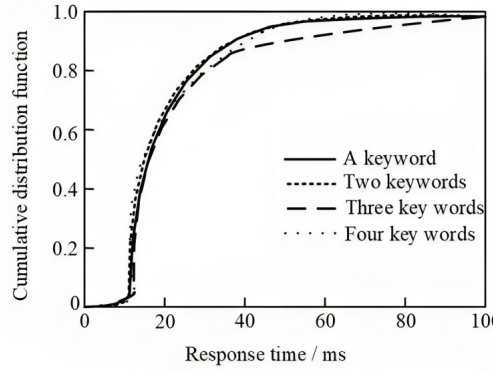
In this paper, by using Python language tools, combined with the relevant resource content existing in Python language, the improved Python based scratch crawler crawls the recruitment data existing in the Internet of Things database, extracts the recruitment data keywords through the text space vector model, mines the recruitment data through the time series model that eliminates the characteristic



a) Reference [3] method cluster mining response time



b) Reference [5] method cluster mining response time



c) Paper method cluster mining response time

Figure 9. Comparison curves of numerical solutions at different times

delay, and finally realizes clustering mining through the clustering algorithm. Finally, taking the data mining of recruitment website as an experimental case, the effectiveness of this method is verified. This method can continue to be optimized in the future, so that the Internet of Things database data can be positioned and analyzed more accurately.

6 DECLARATIONS

Funding: Science and Technology Research Project of Jiangxi Provincial Department of Education: Research on Clustering Mining Algorithm of Internet of Things Database Based on Python Language (Project No. GJJ218415).

Conflict of Interest: There is no conflict of interest among the authors.

Data Availability: All data generated or analyzed during this study are included in the manuscript.

Code Availability: Not applicable.

Author's Contributions: Fang Wan and Ying Liu contributed to the design and methodology of this study, the assessment of the outcomes and the writing of the manuscript.

REFERENCES

[1] Xi, J.—Tang, J. L.: Research on Task Slot Scheduling of Internet of Things Terminal Based on Dynamic Topology. Computer Simulation, Vol. 39, 2022, No. 1, pp. 366–369 (in Chinese).

[2] Vandebon, J.—Coutinho, J. G. F.—Luk, W.—Nurvitadhi, E.: Enhancing High-Level Synthesis Using a Meta-Programming Approach. IEEE Transactions on Computers, Vol. 70, 2021, No. 12, pp. 2043–2055, doi: 10.1109/TC.2021.3096429.

[3] Zhu, A.: Spatiotemporal Feature Mining Algorithm Based on Multiple Minimum Supports of Pattern Growth in Internet of Things. The Journal of Supercomputing, Vol. 76, 2020, No. 12, pp. 9755–9771, doi: 10.1007/s11227-020-03217-x.

[4] Amini, S. M.—Karimi, A.: Two-Level Distributed Clustering Routing Algorithm Based on Unequal Clusters for Large-Scale Internet of Things Networks. The Journal of Supercomputing, Vol. 76, 2020, pp. 2158–2190, doi: 10.1007/s11227-019-03067-2.

[5] Zhou, Q.—Hao, J. K.—Wu, Q.: Responsive Threshold Search Based Memetic Algorithm for Balanced Minimum Sum-of-Squares Clustering. Information Sciences, Vol. 569, 2021, pp. 184–204, doi: 10.1016/j.ins.2021.04.014.

[6] Castellanos-García, L. J.—Elci, S. G.—Vachet, R. W.: Reconstruction, Analysis, and Segmentation of LA-ICP-MS Imaging Data Using Python for the Identification of Sub-Organ Regions in Tissues. Analyst, Vol. 145, 2020, No. 10, pp. 3705–3712, doi: 10.1039/C9AN02472G.

[7] Kitson, E.—Kew, W.—Ding, W.—Bell, N. G. A.: PyKrev: A Python Library for the Analysis of Complex Mixture FT-MS Data. Journal of the American Society for Mass Spectrometry, Vol. 32, 2021, No. 5, pp. 1263–1267, doi: 10.1021/jasms.1c00064.

[8] Schmartz, G. P.—Hartung, A.—Hirsch, P.—Kern, F.—Fehlmann, T.—Müller, R.—Keller, A.: PLSDB: Advancing a Comprehensive Database of Bacterial Plasmids. Nucleic Acids Research, Vol. 50, 2022, No. D1, pp. D273–D278, doi: 10.1093/nar/gkab1111.

[9] Karpachevskiy, A.—Titov, G.—Filippova, O.: Development of a Spatiotemporal Database for Evolution Analysis of the Moscow Backbone Power Grid. Data, Vol. 6, 2021, No. 12, Art. No. 127, doi: 10.3390/data6120127.

[10] Halbersberg, D.—Wienreb, M.—Lerner, B.: Joint Maximization of Accuracy and Information for Learning the Structure of a Bayesian Network Classifier. Machine Learning, Vol. 109, 2020, No. 5, pp. 1039–1099, doi: 10.1007/s10994-020-05869-5.

[11] Manogaran, G.—Srivastava, G.—Muthu, B. A.—Baskar, S.—Shakeel, P. M.—Hsu, C. H.—Bashir, A. K.—Kumar, P. M.: A Response-Aware Traffic Offloading Scheme Using Regression Machine Learning for User-Centric Large-Scale Internet of Things. IEEE Internet of Things Journal, Vol. 8, 2020, No. 5, pp. 3360–3368, doi: 10.1109/JIOT.2020.3022322.

[12] Popko, V. V.—Sentemova, D. V.—Yarmalovich, M. A.: Optimized Reproduction, Collection, Transmission, Processing, and Display of Measurement Data Obtained Using the National Standard of Voltage Unit – Volt. Measurement Techniques, Vol. 64, 2021, No. 7, pp. 556–561, doi: 10.1007/s11018-021-01978-2.

[13] Azarafza, M.—Koçkar, M. K.—Faramarzi, L.: Spacing and Block Volume Estimation in Discontinuous Rock Masses Using Image Processing Technique: A Case Study. Environmental Earth Sciences, Vol. 80, 2021, No. 14, Art. No. 471, doi: 10.1007/s12665-021-09768-3.

[14] Aslam, A.—Chen, H.—Jin, H.: Pre-Filtering Based Summarization for Data Partitioning in Distributed Stream Processing. Concurrency and Computation: Practice and Experience, Vol. 33, 2021, No. 20, Art. No. e6338, doi: 10.1002/cpe.6338.

[15] Liu, S.—Ding, F.—Yang, E.: Iterative State and Parameter Estimation Algorithms for Bilinear State-Space Systems by Using the Block Matrix Inversion and the Hierarchical Principle. Nonlinear Dynamics, Vol. 106, 2021, No. 3, pp. 2183–2202, doi: 10.1007/s11071-021-06914-1.

[16] Burns, S.—Collisson, E. A.: Blockchain-Authenticated Sharing of Cancer Patient Genomic and Clinical Outcomes Data. Journal of Clinical Oncology, Vol. 38, 2020, Art. No. e19358, doi: 10.1200/JCO.2020.38.15_suppl.e19358.

[17] Iggena, T.—Bin Ilyas, E.—Fischer, M.—Tönjes, R.—Elsaleh, T.—Rezvani, R.—Pourshahrokhi, N.—Bischof, S.—Fernbach, A.—Parreira, J. X. et al.: IoTCrawler: Challenges and Solutions for Searching the Internet of Things. Sensors, Vol. 21, 2021, No. 5, Art. No. 1559, doi: 10.3390/s21051559.

[18] Kim, K. O.—Jun, B. J.—Lee, B.—Park, S. J.—Roh, G.: Comparison of First Criticality Prediction and Experiment of the Jordan Research and Training Reactor

(JRTR). Nuclear Engineering and Technology, Vol. 52, 2020, No. 1, pp. 14–18, doi: 10.1016/j.net.2019.06.027.

[19] Kislyi, A. A.—Ravkin, Y. S.—Bogomolova, I. N.—Tsybulin, S. M.—Starikov, V. P.: Number and Distribution of the Narrow-Headed Vole Lasiopodomys gregalis (Pallas, 1779) (Cricetidae, Rodentia) in Western Siberia. Biology Bulletin, Vol. 48, 2021, No. 10, pp. 1822–1831, doi: 10.1134/S1062359021100162.

[20] Rivero-Contreras, M.—Engelhardt, P. E.—Saldaña, D.: An Experimental Eye-Tracking Study of Text Adaptation for Readers with Dyslexia: Effects of Visual Support and Word Frequency. Annals of Dyslexia, Vol. 71, 2021, pp. 170–187, doi: 10.1007/s11881-021-00217-1.

[21] Imran, S. S.—Zainab, H.: Degree of Separation Between Articular Facets (Anterior and Middle) on Anterior Third of Superior Surface of the Calcaneum in H.K.E Region. International Journal of Anatomy and Research, Vol. 8, 2020, No. 3.2, pp. 7633–7638, doi: 10.16965/ijar.2020.124.

[22] Wei, J.—Du, W.—Zhang, B.—Cui, J.—Yi, J.—Kang, S.—Li, J.—Feng, C.: Effect of Interval Cooling Time Between Weld Passes on Temperature Field of Inconel 625. Ordnance Material Science and Engineering, Vol. 43, 2020, No. 6, pp. 90–94 (in Chinese).

[23] Sahoo, P.—Roy, I.—Ahlawat, R.—Irtiza, S.—Khan, L.: Potential Diagnosis of COVID-19 from Chest X-Ray and CT Findings Using Semi-Supervised Learning. Physical and Engineering Sciences in Medicine, Vol. 45, 2022, No. 1, pp. 31–42, doi: 10.1007/s13246-021-01075-2.

Fang Wan is a Lecturer. She had received her Bachelor Degree in 2004 from the Nanchang University in the Department of Software Engineering. Then, she was awarded the Master degree in 2008 from the Nanchang University in the Department of Software Engineering. Now she is working in the Nanchang JiaoTong Institute. Her research interests include electronic commerce and data analysis. She has published one academic paper. Meanwhile, she is also responsible for compiling several textbooks. She also participated in two cooperative research projects.

Ying Liu is an Associate Professor. She received her Bachelor Degree in 2004 from the Nanchang University in the Department of Software Engineering. Now she is working in the Nanchang JiaoTong Institute. Her research areas include database and computer graphics. She has published two academic papers. Meanwhile, she also participated in science and technology project of the Jiangxi Provincial Department of Education.

Computing and Informatics, Vol. 42, 2023, 1158–1183, doi: 10.31577/cai_2023_5_1158

MULTI-AGENT DYNAMIC LEADER-FOLLOWER PATH PLANNING APPLIED TO THE MULTI-PURSUER MULTI-EVADER GAME

Mohammed El Habib Souidi, Makhlouf Ledmi,
Toufik Messaoud Maarouk, Abderrahim Siam,
Abdeldjalil Ledmi

Department of Computer Science, ICOSI Lab, University of Khenchela, 40004 Khenchela, Algeria
e-mail: souidi.mohammed@univ-khenchela.dz

Abstract. Multi-agent collaborative path planning focuses on how the agents have to coordinate their displacements in the environment to achieve different targets or to cover a specific zone in a minimum of time. Reinforcement learning is often used to control the agents' trajectories in the case of static or dynamic targets. In this paper, we propose a multi-agent collaborative path planning based on reinforcement learning and leader-follower principles. The main objectives of this work are the development of an applicable motion planning in a partially observable environment, and also, to improve the agents' cooperation level during the tasks' execution via the creation of a dynamic hierarchy in the pursuit groups. This dynamic hierarchy is reflected by the possibility of reattributing the roles of Leaders and Followers at each iteration in the case of mobile agents to decrease the task's execution time. The proposed approach is applied to the Multi-Pursuer Multi-Evader game in comparison with recently proposed path planning algorithms dealing with the same problem. The simulation results reflect how this approach improves the pursuit capturing time and the payoff acquisition during the pursuit.

Keywords: Multi-agent system, path planning, pursuit-evasion game, reinforcement learning

Mathematics Subject Classification 2010: 68T40-68T42

1 INTRODUCTION

Multi-agent Pursuit-Evasion Game (PEG) can be considered a multi-task problem in which different groups of pursuers' agents are trying to block the motion of another group of detected evaders' agents [1]. PEG is usually processed through the utilization of a task coordination mechanism and a path planning method. On the one hand, the task coordination mechanism [2] is used to allow an efficient formation of different multi-agent coalitions able to execute the multi-task problem. On the other hand, the path planning method [3] allows the pursuers' agents to trace the trajectories leading them to the evaders' positions.

Multi-agent organizational models [4] are considered as a type of multi-agent task coordination mechanisms. An organizational model can be defined as a metamodel reflecting the relations between the concepts used simultaneously to coordinate the collective behavior of the agents. For example, in [5] the authors used the concepts of Agent, Group, and Role and the relations between them to propose a multi-agent organizational metamodel.

In the recent research activities, MAS organizational modeling frameworks [6] are enormously used to the coordinate the tasks in the PEG. Recently in [7], the authors have based on the different concepts forming the Yet Another Multi-Agent Model (YAMAM) [8] to create an efficient pursuit groups' access mechanism. Furthermore, supervised and unsupervised machine learning methods [9] are used in combination with organizational models to improve the tasks' coordination. On the one hand in [10], the authors used the neural networks' layer [11] to extract the features of the AGRMF model. To allow the coalition of the pursuers with similar features, the extracted features are processed via a self-organizing map layer. On the other hand, in [12], the authors used K-means [13] in order to group the similar evaders characterized by the best parameters among the data set.

Multi-agent collaborative path planning can be defined as the generation of a continuous series of movements from the initial to the final state of each agent, while at the same time avoiding collisions with the other agents. Markov Decision Process (MDP) [14] is a stochastic process usually used in MAS with the aim of modeling the path environment. This modeling allows the agent to make the decision according to several possible transitions. The main goal of reinforcement learning [15] is to provide the agent with an intelligent behavior during the movements through the optimization of the expected payoffs. However, in decentralized multi-agent path planning, each agent moves without taking into consideration the behavior of other agents. In addition, the multi-agent path planning problem is processed via the use of different optimization methods such as Genetic Algorithm [16, 17], Particle Swarm Optimization (PSO) [18, 19], as well as Artificial Potential Field [20, 21].

MAS collaborative path planning is both important and challenging for several reasons. The first objective of MAS path planning is to guarantee that agents can displace without colliding with each other or with the environment's obstacles. In complex environments with multiple dynamic agents such as the PE game environ-

ment, finding collision-free paths for all agents becomes increasingly hard. Thus, collision avoidance can be considered crucial to ensure the safety as well as the integrity of the agents and their environment. Moreover, MAS path planning involves the coordination and cooperation of the agents with each other with the aim of achieving common objectives. This fact requires the agents to consider the actions and the intentions of the other agents, and to predict their future behavior. Consequently, coordinating MAS efficiently is a challenging task, knowing that the agents have to balance their individual objectives with the overall system's goals. Furthermore, we can easily note that real-world scenarios usually involve uncertainty as well as dynamic changes in the environment. In some cases, agents have incomplete information about the positions, velocities, or intentions of the other agents. In addition, the environment itself may change over time due to dynamic obstacles, unpredictable events, or varying objectives. Therefore, the incorporation of uncertainty and adaptability into MAS path planning process further increases the complexity of the problem.

In this paper, we introduce a new cooperative multi-agent path planning through its application to the PEG. This method is based on a dynamic attribution of the sub-roles Leader and Followers to the pursuers belonging to the same group according to their dynamic environmental positions to decrease the pursuit capturing time. Knowing that the environment is modeled according to MDP principles, the hierarchy of each pursuit group is dynamically updated in relation to the rewards detected by the pursuers in the pursuit environment. The main contributions of this paper can be summarized as follows:

- The proposition of a MAS collaborative path planning based on the hierarchization of the agents in dynamic roles as well as Q-learning to orient the agents in selecting their directions.
- The applicability of the proposed approach in a partial observable environment. Knowing that the followers are not required to know the targets' positions in order to perform their tasks.
- The application of the proposed approach to the Multi-Pursuer Multi-Evader Game (MPMEG) in comparison with recent path planning methods. During this study, we have taken into account the capturing time as well as the pursuers' development during the game execution in order to prove the feasibility of the proposed approach.

The paper is organized as follows: In Section 2, we discuss the main related works processing the multi-agent coordination as well as the PE problems in relation to the proposed work. In Section 3, we describe the PE game environment via the definition of its different components. Furthermore, we explain the difference between the agents existing in the environment, and how this last is modeled according to the MDP principles. Section 4 details how the PE game is processed through the application of the new path planning proposed in this paper. Section 5 reflects the simulation results obtained in comparison with a recent approach also

dealing with PE problem. Finally, Section 6 highlights concluding points regarding the proposed approach.

2 RELATED WORK

Among the recent and interesting works regarding the multi-agent path planning in PEG, we note [22], in which the authors proposed a static leader-follower path planning based on reinforcement learning. This work is based on the decomposition of tasks between the pursuers. In other words, each pursuit group is composed of a set of Leaders and a set of Followers. In comparison with this path planning algorithm, our proposal is based on the attribution of only one role Leader per each pursuit group to focalize the pursuers on the global best solution. Moreover, in this work, the roles Leader and Followers of the moving pursuers belonging to each group are dynamically reattributed before each pursuit iteration according to the agents' new positions. Knowing that the dynamic reattribution of these roles positively impacts the pursuit capturing time as well as the pursuers' development in case of moving evaders.

In [23], the authors processed the PEG through the processing of the constraints linked to the environment changes during the pursuit. They proposed a deep reinforcement learning method allowing the capture of the evaders even if the number of pursuers has changed. Specifically, they have based on a deep deterministic policy gradient (DDPG) framework and bi-directional recurrent neural network (Bi-RNN) with the aim of studying the PEG in the case where the evaders are faster than the pursuers, but less numerous than them. However, the authors did not introduce a task allocation method to define which pursuers must perform the pursuit of specific evaders in the case of MPMEG. This fact negatively impacts the autonomy of the approach and its application in the real world. To overcome this limitation in this proposed approach, we have based on the MAS task coordination mechanism based on YAMAM organizational model proposed in [7].

In [7], the authors used Q-learning [24] in order to allow the pursuers to move in a decentralized way with the aim of obtaining the maximum payoffs detected in environment cells. Knowing that the payoffs are calculated in relation to the distance separating each environment cell from the cell containing the concerned evader. However, the negative point of this approach is reflected by the fact that it is only applicable in a completely observable environment. In other words, all the pursuers need to know the exact position of the concerned evader at each pursuit iteration. However, in the proposed approach, the pursuit can be applied in a partially observable environment. In other terms, only the leader needs to know the exact position of the concerned evader.

In [25], the smart pursuers undertook Watkins's Q(λ)-learning algorithm with the aim of learning from their interactions. The method the authors used is an extended version of Q-learning and eligibility traces. It utilized saved knowledge until the first occurrence of an exploration. We can note that in the two last related

works [24, 25] the authors explained that each pursuer takes independent decisions regarding its action-value function and the updates of its information space. In comparison with the proposed approach in this paper, the path planning proposed in [25] only processed the Multi-Pursuer Single-Evader Game (MPSEG), which is a less complex problem in relation to the MPMEG processed in this paper.

In [26], the authors have based on game theoretic principles and Q-learning to process the PE problem. After the formation of the hunting team and via learning from the evader's path strategy, the trajectory of the evader's limited T-step cumulative payoff is generated and adjusted to the pursuer's strategy set. Also, the game theoretic Nash equilibrium solution is obtained through the resolution of the cooperative pursuit game. finally, each pursuer follows the generated equilibrium strategy to complete the pursuit task. However, this approach is based on a centralized communication method with several lacks. In other words, this approach totally depends on the virtual manager which identifies the pursuers and the evader, records the agents' paths, and selects the best solution in a centralized way in the case of finding several balanced solutions. Consequently, we can conclude that the approach is not realizable in the case where the virtual manager is out of order. In comparison with the proposed approach, the PE game processing is more distributed on the integrality of the pursuers. In other words, in case of the leader's failure, this last is immediately replaced by the pursuer with the highest pursuit skills, which will be introduced in the next section of this paper.

PEG can even be considered a clearing zone problem where the pursuers are trying to cover the pursuit environment in a minimum time to detect the evaders' positions. In [27], a Partially Observable Markov Decision Process (POMDP) algorithm is illustrated to localize the mobile target in a known graph. The main objective is to perform the capture of the targets through the clearing of the graph as quickly as possible. Otherwise, this approach is clearly limited by the pursuers' field of view as well as the camera type used in order to detect the evaders. In relation to the proposed work, the approach proposed in [27] is not based on Q-learning principles. Moreover, it processed the Single-Pursuer Single-Evader game. Finally, the PE game processed in [27] is not situated in the grid of cells environment.

The PEG is also processed via the avoidance of the different obstacles detected in the environment. In [28], the authors proposed a new obstacle avoidance path planning-based MDP framework and bug algorithms [29]. Knowing that the main objective of the bug algorithm is to unidirectionally turn around the obstacle until finding the leaving point, they proposed to find the leaving point according to the payoffs returned by the application of the MDP reward function. This approach provided interesting results in relation to the precedent bug algorithms, however, this approach requires an MDP environment modeling in order to be applicable. In the PE game, path planning algorithms and obstacle avoidance methods must be combined to allow the agents' displacement in a pursuit environment that contains obstacles. Thus, in the case of a pursuit environment with obstacles, the proposed path planning in this paper can be combined with the obstacle avoidance algorithm proposed in [28] to provide an efficient pursuers' behavior.

Regarding the task sharing between the pursuers, in [7] the authors based on the concepts of Agent, Role, Task, and Skills forming the YAMAM organizational model to develop a pursuit groups access mechanism allowing an equitable and stable grouping of the pursuers during the pursuits. In [30], they proposed a coalition formation algorithm for the student agents selecting the courses proposed by the university in E-learning system. Precisely, they introduced a voting procedure allowing the coalition of the agents and also the allocation points of the different courses. The proposed method makes the agents able to independently express their preferences and simultaneously use the information furnished by the precedent rounds to vote intelligently and strategically. In MPMEG, in addition to the path planning, a task-sharing method should also be provided to determine which pursuers have to pursue each evader. In this paper, we have based on the task-sharing method proposed in [7] to allow the pursuit groups' formation.

3 PROBLEM DESCRIPTION

PEG is considered a multi-agent complex problem in which moving agents known as Pursuers are forming different pursuit groups in order to capture other moving agents known as Evaders. The main objective of this game is to decrease the pursuers displacements during the pursuit by providing them an intelligent behavior. On the one hand, this behavior is usually reflected by the use of an intelligent task coordination algorithm allowing the pursuers to be regrouped in different pursuit groups according to their abilities. On the other hand, it is reflected by the use of collaborative path planning algorithms that provides to each pursuer an optimal trajectory to follow in order to achieve the goal location.

In this work, the PE game will be handled in a limited grid of cells environment, in which the agents can displace from a cell to another one according to the detected information as well as to their velocities. The agents can indirectly communicate through the modification of environment information. Each cell is characterized by Cartesian coordinates and also a vector containing different information. This set of information represents the expected payoff that could be obtained by the pursuers in the case of reaching the concerned cell. Moreover, it reflects the information regarding the cell occupation. Knowing that each cell can only contain one agent at a time. Moreover, the rewards contained in each cell are dynamically updated during each pursuit iteration according to the new positions of the moving evaders.

In this proposal, there exist two types of agents in the environment, the evaders and the pursuers in accordance with the PEG principles. These agents can move in four different directions: up, down, left, and right. Moreover, they are equipped with limited sensors allowing them to read the information contained in the adjacent cells. Knowing that the main objective of the PE game is to stop the movement of the detected evaders as quickly as possible, the evaders are randomly moving in the environment with the aim of avoiding their captures. Furthermore, each evader

needs the coalition or a pursuit group formed by a specific number of pursuers in order to be captured.

With the aim of parallelly performing the pursuit of the detected evaders, the pursuit group regarding each evader must contain a specific number of pursuers according to the evader's requirement. In addition, the pursuers must follow a specific collaborative path planning in a centralized or decentralized way in order to move in the direction of these evaders. The centralized path planning concerns the case where the pursuers are collaborating during their displacements. However, the decentralized way regards the case where the pursuers are independently moving in the direction of the goal cell. To propose a new collaborative path planning in this paper, the PE environment is modeled as an MDP framework (S, A, R, T):

- $S = \{s_1, s_2, \ldots, s_n\}$: the set of the environment states (cells). In the PE game, the states represent the cells existing in the environment.
- $A = \{a_1, a_2, \ldots, a_n\}$: the set of actions that the agent can effectuate. In the PE game, the actions can be described by the displacement of the agent in the environment, and also by the reading of the information contained in the adjacent cells.
- $R(s, a)$: The reward function determines the payoff could be obtained by an agent if it reaches the state s through the execution of the action a in the PE game, the payoff is proportionally inverse to the distance between the cell containing a pursuer and the cell containing the concerned target.
- $T(s, a, s')$: The transition function determines the impact of an agent's action on the environment. It also determines the probability of switching from the state s to s' by executing the action a according to the payoff returned by the two states.

To allow the pursuers' collaboration, we propose to introduce two types of pursuers, the leader and the followers. The leader of a pursuit group is the detector of the concerned evader's position. In other words, the leader of a pursuit group is the pursuer occupying the closest position in relation to the concerned evader's position. The role of the leader is to guide the followers belonging to the same group to capture the evader via the modification of the environmental information. The followers must minutely move according to the path traced by the leader. The Q-learning in this proposal is used to guide the leaders to move in the direction of the pursuers and also to allow the followers to move according to the desired path traced by the leaders.

Figure 1 is a part of the pursuit environment taken from the effectuated simulations in which 2 evaders are pursued by two pursuit groups formed by 4 pursuers each. The variables of the vector [Var_1, Var_2, Var_3, Var_4, Var_5] contained in each cell are explained in the following way:

- Var_1: the reward returned to the leader of the first pursuit group in the case of reaching the concerned cell.

- Var_2: the reward returned to the leader of the second pursuit group in the case of reaching the concerned cell.
- Var_3: the cell index, it can contain 2 different values:

$$Index\_cell = \begin{cases} 1, & \text{if } (\exists Ag \in agents\_list \wedge Ag \subset cell) \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

- Var_4: the reward returned to the followers of the first pursuit group in the case of reaching the concerned cell.
- Var_5: the reward returned to the followers of the second pursuit group in the case of reaching the concerned cell.

Furthermore, this figure showcases 5 dispersed agents having different colours. The green agent represents one of the detected evaders. The yellow agent represents a pursuer and at the same time the leader of the pursuit group trying to capture the green evader. The red agents are pursuers and at the same time followers of the yellow leader. The different payoffs of each cell are calculated in relation to the distance between the evader and the pursuers according to their types.

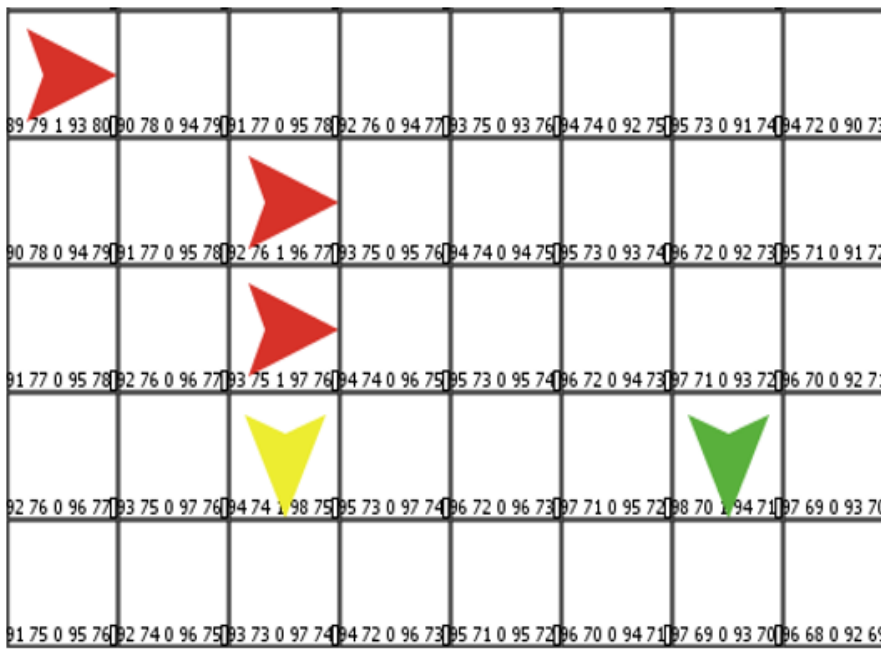


Figure 1. PEG environment's part

4 LEADER-FOLLOWER PATH PLANNING

In this section, we introduce the proposed path planning algorithm. Firstly, we explain how this algorithm can be generally used. Secondly, we detail how the algorithm is applied in order to provide the pursuers' trajectories during the PEG processing.

Algorithm 1 details how the leader-follower principle is used in order to manage the multi-agent path planning until the end of the task execution. The agent's skill can be defined as different ability factors of the agent that play an important role during the task execution. For example, in the case of the PEG, the environment position and velocity of each pursuer are considered as important factors with a high impact on the pursuit processing. The agent's skill is generally calculated in relation to agent's ability factors as follows:

$$\lambda(Ag_i) = \Omega_1 * ab_1^i + \Omega_2 * ab_2^i + \cdots + \Omega_n * ab_n^i, \tag{2}$$

- ab_k^i: the k^{th} ability factor of the agent Ag_i,
- Ω_i: the i^{th} ability factor's coefficient of the agent, $\Omega_i \in [0, 1]$.

We note that the agents' skill calculation is updated before every new iteration with the aim of reattributing the dynamic role of Follower to the best agent belonging to the group. The desired trajectory can be defined as the path taken by the leader. The desired trajectory is updated according to the leader's last position. In MDP environment, this update is performed through the modification of followers' expected payoffs in the environment, as shown in Figure 1. The role of the followers is to move according to the desired trajectory traced by the leader. The *leader-index* variable shown in Algorithm 1 allows the attribution of the role Leader to only one agent.

Algorithm 2 details how the leader-follower path planning is used in order to solve the PE game. The first step of the PE game is the detection of Cartesian coordinates of the existing evaders. To do this, each agent scans a specific surface of the environment. A pursuit group for each evader will be created to stop the movement of this last. With the aim of optimizing the access to the created pursuit groups, we have based on a recent task coordination mechanism based on Yet Another Multi-Agent Model (YAMAM) [7]. This organizational modeling framework is based on 4 concepts: Agent, Role, Task, and Skills. In order to participate in the pursuit achievement (task), the agent must obtain the role of Pursuer proposed by a specific group. To get this role, the concerned agent must be characterized by specific skills' degrees.

After the pursuers' roles attribution, the leader and the followers of each pursuit group will be designated. Knowing that in each pursuit group only one pursuer can obtain the role of Leader. In order to play this role, the pursuer must be characterized by the highest pursuer' skill degree. In relation to PEG, this degree is calculated in relation to the reward of the pursuer's position, the pursuer's velocity,

Input: task-execution = false;
Agents-group-initialization();
while *task-execution() = false* **do**
  leader-index ⟵ false;
  Agents-skills-calculation();
  for *each* $agent_i$ **do**
    if $agent_i - skills = \max \wedge$ *leader-index = false* **then**
      Role-attribution($agent_i$, Leader);
      leader-index ⟵ true;
    end
  end
  Leader-action();
  Update(desired-trajectory());
  for *each* $agent_i$ **do**
    if $agent_i \neq Leader$ **then**
      Role-attribution($agent_i$, Follower);
      Follow-desired-trajectory($agent_i$);
    end
  end
  Check(task-execution());
end
Output: task-execution = true;

Algorithm 1: Multi-agent leader-follower path planning

as well as the sensor's length of the pursuer. The agent's skill is calculated as follows:

$$\lambda(P_i) = \alpha * r_i + \beta * v_i + \gamma * SL_i, \tag{3}$$

where:

- r_i: the reward of the pursuer_i according to its position in the environment;
- v_i: the agent's velocity determining the number of cells that the pursuer_i can cross during an iteration;
- SL_i: represents the average sensor's length of the pursuer_i.
- α: the reward coefficient, $\alpha \in [0, 1]$;
- β: the velocity coefficient, $\beta \in [0, 1]$;
- γ: the Sensor's Length coefficient, $\gamma \in [0, 1]$.

Knowing that the pursuer can move in 4 different directions, the average Sensor's Length is calculated as follows:

$$SL_i = \gamma_1 * SL_i^{up} + \gamma_2 * SL_i^{down} + \gamma_3 * SL_i^{left} + \gamma_4 * SL_i^{right}. \tag{4}$$

In the case where the role of Leader is already attributed, the pursuer automatically obtains the role Follower. The role Leader consists on moving forward in the direction of the evader according to the leader's payoff shown in Figure 1. After the leader movement, the followers expected payoffs are updated. The role of the followers is to move forward in the direction of their leader according to the followers' payoff shown in Figure 1.

Knowing that the environment is modeled according to MDP principles, the Q-learning algorithm can be placed within MDP framework with the aim of allowing the pursuers to move via the learning of the optimal Q-values defined in Equation (5). This reinforcement learning method allows to learn a strategy, which indicates what action to perform in each state of the system. It works by learning a state-action value function denoted Q which makes it possible to determine the potential gain. In other words, in Q-learning, each pursuer executes an action (a) in relation to the state (s) and to the function Q. The pursuer then perceives the new state (s') and a reward (r) from the environment before the update of the Q function.

$$Q^*(s,a) = R(s,a) + \Lambda \sum_{s'} T(s'|s,a)V^*(s'), \tag{5}$$

where:

$$V^*(s) = \max_a Q^*(s,a) \tag{6}$$

and

- Λ: The discount factor ($\Lambda \in [0,1]$).

During every iteration, the pursuers takes the decision (moving up, down, right, or left) that maximizes their payoff according to the equation below:

$$Q_{i+1}(s,a) = Q_i(s,a) + \alpha_i * [r_{i+1} + \Lambda * \max_{a'}(Q_i(s_{Up},a'), Q_i(s_{Down},a'), Q_i(s_{Left},a'), Q_i(s_{Right},a')) - Q_i(s,a)], \tag{7}$$

where

- α_i: The step-size sequence.

In the case of a leader path planning, the reward r of the Equation (7) is calculated in relation to the distance separating the leader from the concerned evader as follows:

$$r_{Leader} = r_{max} - \sqrt{(CC_{Leader_x} - CC_{Evader_x})^2 + (CC_{Leader_y} - CC_{Evader_y})^2}. \tag{8}$$

In the case of a follower path planning, the reward r is calculated in relation to the distance separating the follower from the leader of the pursuit group they belong to:

$$r_{Follower} = r_{Leader} - \sqrt{(CC_{Leader_x} - CC_{Follower_x})^2 + (CC_{Leader_y} - CC_{Follower_y})^2}, \tag{9}$$

where:

- r_{max}: the reward could be returned to the pursuer in the case of reaching one of the concerned evader's adjacent cells;
- $(CC_{Leader_x}, CC_{Leader_y})$: Leader Cartesian coordinates;
- $(CC_{Follower_x}, CC_{Follower_y})$: Follower Cartesian coordinates;
- $(CC_{Evader_x}, CC_{Evader_y})$: Evader Cartesian coordinates.

Knowing that before the displacement of the pursuers, the evaders randomly move (random direction) in the environment according to their velocities, as shown in Algorithm 2. The pursuit iteration can be defined as the execution of a possible transition by each agent. In other words, it regards the displacement of the agents from their actual cells to one of their adjacent cells. Before any pursuit iteration, the roles Leader and Followers of each pursuit group are updated to optimize the capturing time and the payoff acquisition. This update is due to the fact that the pursued evaders change their positions in the environment during each pursuit iteration by moving from one cell to another according to a specific velocity. This update is effectuated according to the dynamic distance separating each pursuer from the new position of the concerned evader.

The PEG is considered over when the average payoff obtained by the pursuers reaches a specific value. This value is represented by the variable *max-payoff* in Algorithm 2. Max-reward can be initialized by a value related to the studied case. The index of the pursuit is verified as follows:

$$IC = \begin{cases} \text{True}, & \text{if } \left(\frac{r_{P_1} + r_{P_2} + \cdots + r_{P_{np}}}{np} = r_{max} \right), \\ \text{False}, & \text{otherwise}, \end{cases} \tag{10}$$

where:

- IC: the index of the pursuit capture;
- np: the number of existing pursuers.

Scalability in a MAS refers to the system's ability to manage an increasing number of agents without significant degradation in performance or efficiency. It involves designing the system in a way that allows it to accommodate larger agent populations and more complex interactions without sacrificing its functionality or responsiveness. As shown in Algorithm 2, each pursuer calculates its pursuit skill independently of the other pursuers. This fact proves that the proposed algorithm is based on a decentralized calculation during the selection of the groups' leaders. Therefore, we can conclude that the increase of the agents' number does not negatively impact the algorithm performance.

5 SIMULATION RESULTS

This part summarizes the main simulations performed to showcase the efficiency and also, to prove the feasibility of the approach proposed in this paper. To do

Input: Index-capture = false;
Evaders-detection();
Broadcast(Evaders-coordinates);
Pursuit-groups-formation();
Distance-calculation();
for *each Agent*$_i$ **do**
  Role-Atrribution (Agent$_i$);
end
Initialize(max-payoff);
while *Index-capture* = *False* **do**
  $R \longleftarrow 0$;
  Evaders-move(random);
  Update(r_{leader});
  for *each Group*$_k$ **do**
    for *each* P_i **do**
      if $P_i \in Group_k = true$ **then**
        Distance-calculation (P_i , E_k);
        $\lambda(P_i)$;
        list$_k[] \longleftarrow$ add($\lambda(P_i)$);
      end
    end
    for *each* P_i **do**
      if *condi*$\lambda(P_i) = max(list_k[])$ **then**
        Leader$_k \longleftarrow P_i$;
        Move-to(P_i, E_k);
        Update($r_{follower}$);
      else
        Follower$_k[] \longleftarrow$ add(P_i);
      end
    end
  end
  for *each Group*$_k$ **do**
    for *each* P_i **do**
      if $P_i \in Follower_k[] = true$ **then**
        Move-to($P_i, Leader_k$);
      end
      $R \longleftarrow R + r_{P_i}$;
    end
  end
  if $\frac{R}{np}$ = *max-payoff* **then**
    Index-capture $\longleftarrow$ true;
  end
end
Output: Index-capture = true;

Algorithm 2: Pursuit-evasion based on leader-follower path planning

these, we have used NetLogo 5.0.4 [31], which is an open-source oriented agent platform. It is based on two kinds of agents: the patches and turtles. The patches are situated agents which can stock dynamic information. We have used this type of agent with the aim of implementing our grid of cells environment. Specifically, each patch represents a cell containing the dynamic payoffs. The turtles are the mobile agents which can move from one patch to another. This second type is used to simulate the behaviour of our pursuers (leaders and followers) as well as our evaders. The experiments will be handled in a limitary 100×100 grid of cells environment, where 8 pursuers are trying to capture 2 evaders. Knowing that each evader requires a pursuit group formed by 4 pursuers to be captured. Regarding, the agents' sensor length (SL), we have equipped each agent with sensors able to obtain the information contained in each adjacent cell (Up_{cell}, Down_{cell}, Left_{cell}, Right_{cell}). The initial agents' positions are detailed in Table 1.

Agent	Initial Cartesian Coordinates
P_1	(50, 75)
P_2	(75, 75)
P_3	(25, 50)
P_4	(50, 50)
P_5	(75, 50)
P_6	(37, 38)
P_7	(62, 37)
P_8	(50, 25)
E_1	(55, 55)
E_2	(44, 44)

Table 1. The agents' initial positions

In order to focalize the simulation studies only on the impact of the path planning on the PEG, we have based on the following pursuit groups generated through the application of the task coordination mechanism proposed in [7] with the aim of capturing the mobile evaders E_1 and E_2:

$$\textit{Pursuit-Group}(1) = \{P_4, P_5, P_1, P_2\},$$

$$\textit{Pursuit-Group}(2) = \{P_6, P_3, P_8, P_7\}.$$

In order to showcase the improvement brought by the proposed path planning, we have seen the usefulness of comparing it with two recent path planning methods treating the PE game. The main difference between the compared cases is detailed as follows:

Case A: Pursuit-evasion game based on the new leader-follower path planning explained in Section 3.

Case B: Pursuit-evasion game based on the static leader-follower path planning proposed in [22] and detailed in Section 2 of this paper.

Case C: Pursuit-evasion game based on the path planning proposed in [7], where the pursuers are moving independently of each other according to the dynamic rewards detected in the environment. Specifically, there are no specific roles (Leader and Followers) regarding the pursuers. Each pursuer leads itself to perform the pursuit.

We have seen the usefulness of comparing it with the proposed approach to the cases B [22] and C [7] for the following reasons:

- In relation to the proposed approach in this paper, cases B and C are also based on reinforcement learning principles,
- Cases B and C were recently applied to the MPMEG, which is the processed PE game in this paper.
- These two approaches can also be applied in a grid of cells PE game environment used in this paper to reflect the impact of the proposed approach.

Figure 2 reflects the pursuit capturing time obtained after 20 pursuit episodes. A pursuit episode starts by the coalition formation of the pursuit groups, and ends after the capture of the different evaders. The average capturing time in the case A decreases by 9.91 % in relation to case B and 16.41 % in relation to case C. This fact is totally due to efficiency of the leader-follower proposed technic. In other words, the dynamic attribution of the sub-roles of Leader and Follower increases the goal orientation of the pursuers.

In order to prove the significance of the obtained capturing time reflected in Figure 2, we have performed the Friedman test. According to the obtained results ($X_r^2 = 27.925$), we can conclude that the obtained result is significant at $p < 0.05$ (significance level).

Figure 3 reflects the average reward obtained per iteration by the pursuers during a complete pursuit episode in the 3 compared cases. From this figure, we can note that the average reward increases in case A by 7.1 % in relation to case B, and 13.5 % in relation to case C. We can justify the flagrant difference between case A and C by the close grouping as well as the close displacement of the pursuers during the pursuit provided by the leader-follower principle. The average reward obtained per iteration (Ar_LD) in the cases A and B is calculated as follows:

$$Ar_{LD} = \sum_{i=1}^{nl} \left(r^t_{Leader_i} - r^{t-1}_{Leader_i} \right) + \sum_{i=1}^{nf} \left(r^t_{Follower_i} - r^{t-1}_{Follower_i} \right), \tag{11}$$

where

- t: the index of the pursuit iteration;
- nl: represents the number of leaders used in the game. In these simulations, we are using 2 leaders;
- nf: represents the number of followers used in the game. In these simulations, we are using 6 followers.

However, in the case of decentralized path planning (case C) in which each pursuer behaves as a leader by independently moving to the target, Ar_D is calculated as follows:

$$Ar_D = \sum_{i=1}^{np} \left(r^t_{Pursuer_i} - r^{t-1}_{Pursuer_i} \right), \tag{12}$$

where

- *np*: represents the number of pursuers used in the game. In these simulations, we are using 8 pursuers.

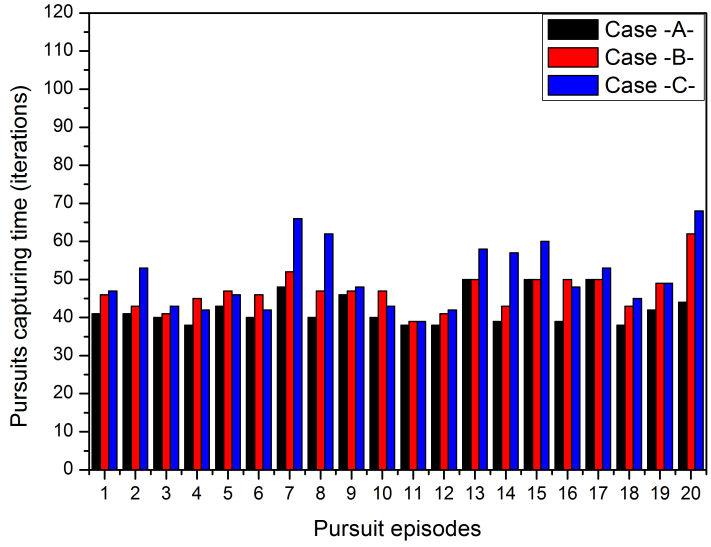


Figure 2. PEG capturing time obtained (Pursuers' motion speed = Evaders' motion speed)

In order to study the behavior of the pursuers during the pursuit, we have focused on their dynamism degree regarding the roles' attribution. This dynamism (Dy) concerns the roles' changes between the roles of Leaders and Followers in the two pursuit groups studied. In other words, if a pursuer changes its role during a pursuit iteration, then the dynamism degree is automatically incremented. It is calculated at each pursuit iteration as follows:

$$\text{For each } P_i, Dy = \begin{cases} Dy, & \text{if role}^{t-1}_{P_i} = \text{role}^{t}_{P_i}, \\ Dy + 1, & \text{otherwise.} \end{cases} \tag{13}$$

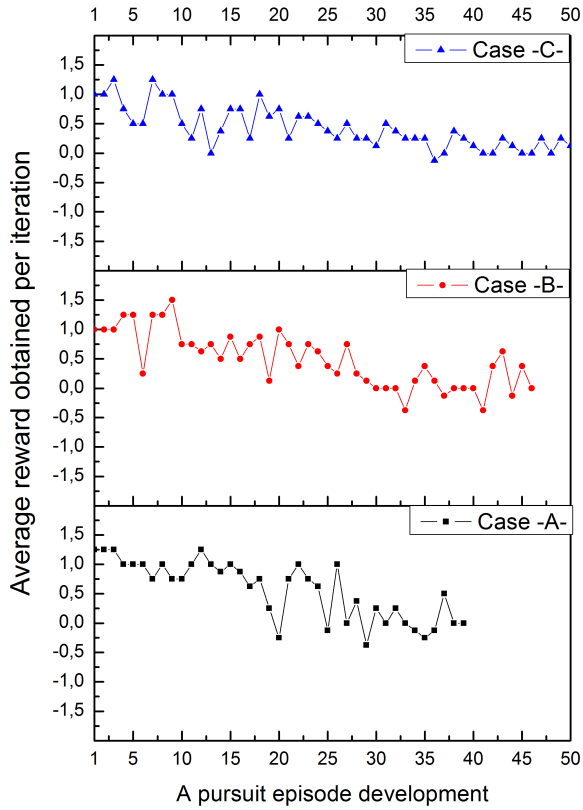


Figure 3. Pursuer's average rewards obtained per pursuit iteration (Pursuers' motion speed = Evaders' motion speed)

As shown in Figure 4, we note 8 roles attributions at the first iteration which can be explained by the creation of 2 pursuit groups in which each group is formed of 4 pursuers. Moreover, we can note that this degree increases in the case when the pursuers are closer to the evaders (from the 27^{th} iteration).

Furthermore, we have studied the dynamism of the role's attributions during 20 pursuit episodes regarding case A, as shown in Figure 5. The average dynamism obtained is 25.65. We can note that this degree is proportionally inverse to the pursuit capturing time. Knowing that the difference between the results obtained in each pursuit episode is related to the fact that the pursuers are randomly moving in the environment. Consequently, we conclude that the PEG processed in these simulations is non-deterministic.

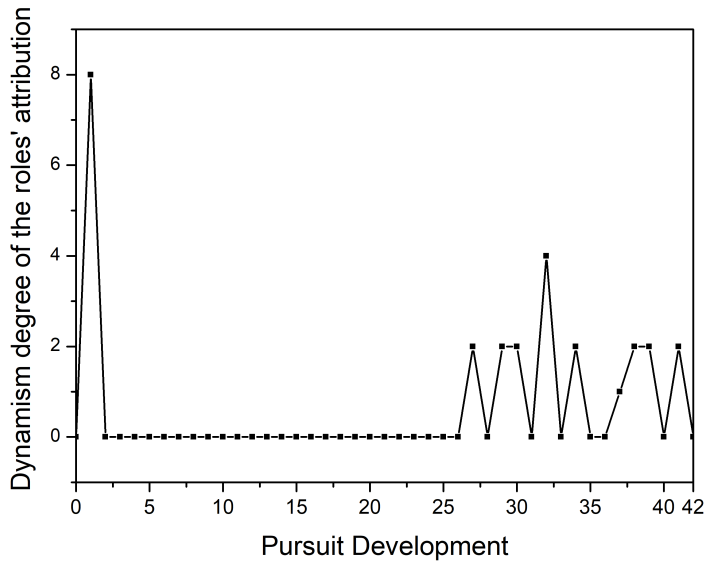


Figure 4. The study of the dynamism degree of the roles' attribution during the pursuit iterations of a complete pursuit episode

With the aim of varying the studied cases, we have doubled the evaders' motion speed in relation to the pursuers. The results shown in Figure 6 showcase the capturing times obtained after 20 pursuit episodes in the both cases. We can easily note that the difference between the three cases increases in relation to the case using the same motion speed (Figure 2). The average capturing time in case A decreases by 18.14 % comparing with case B and 33.81 % comparing with case C. We can conclude that the difference in the capturing time between the two cases increases in relation to the increase of motion speed's difference between the pursuers and evaders.

In accordance to the reflected results in Figure 7, we observe that the maximum average reward is reached after 49 pursuit iterations in case A, 56 iterations in case B, and 76 iterations in case C. In relation to the path planning proposed in case B and C, the dynamic leader-follower approach increases the cooperation level between the pursuers belonging to the same group in order to efficiently accomplish the assigned task.

Moreover, we have also effectuated the Friedman test on the development of the payoff acquisition shown in Figure 7. Knowing that the Friedman test requires the same number of values to be performed, we have only taken into account the results obtained in the first 49 pursuit iterations. According to the obtained result

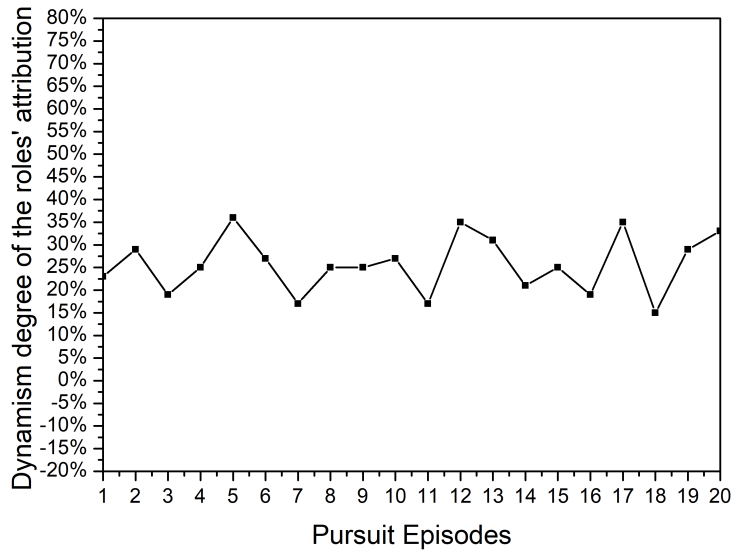


Figure 5. The study of the dynamism degree of the roles' attribution during 20 pursuit episodes

($X_r^2 = 25.5408$), we can conclude that the obtained result is significant at $p < 0.05$.

In order to compare the dynamism of the roles' attribution between the cases with different evaders' velocities, we have also studied this factor during 20 pursuit episodes in which the velocity of the evaders is doubled in comparison with the pursuers' velocity, as shown in Figure 8. From the showcased results, we can note that the average dynamism degree is 39.9. In comparison with the results reflected in Figure 5, we deduce that the dynamism degree increases by 35.71 %. From this result, we can conclude that the dynamism of the roles' attribution increases in relation to the increase of the evaders' velocity. Knowing that when the pursuers change their roles, that means that they find a better strategy in relation to the undertaken strategy during the pursuit iteration $(t-1)$.

Table 2 summarizes the average capturing time obtained after 20 pursuit episodes and also the average reward obtained per iteration in the case where the speed of the pursuers and evaders is the same, and also in the case where the evaders' motion velocity is doubled.

In this section, we have studied the proposed path planning algorithm in comparison with recent approaches [7, 22] dealing with the same problem. During these simulations, we have varied the evaders' velocities to increase the complexity of the pursuit. From the obtained results, we have constated that the pro-

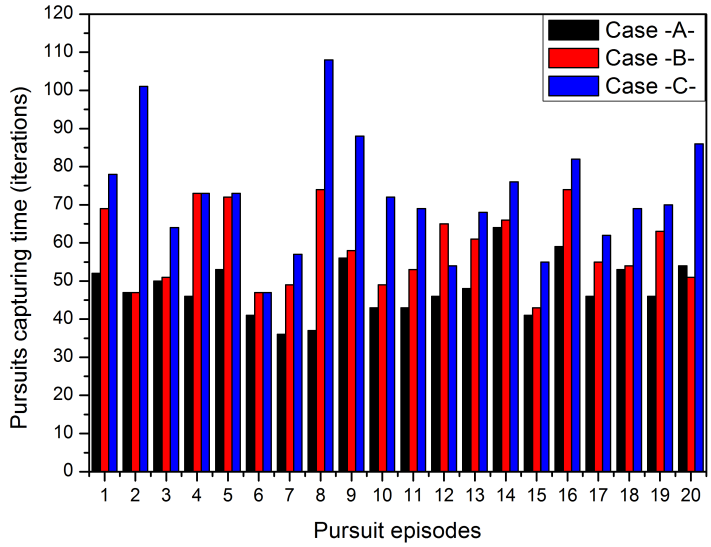


Figure 6. PEG capturing time obtained (2 × pursuers' motion speed = evaders' motion speed)

posed approach improves the capturing time as well as pursuers' reward acquisition in relation to the compared methods. Furthermore, we have noted that the proposed approach is less impacted by the increase of evaders' velocity in comparison to the other works, which is due to the goal-orientation of the novel approach. Also, we have studied the pursuit groups reorganization through the dy-

		Case A	Case B [22]	Case C [7]
Pursuers' motion speed = Evaders' motion speed	Average capturing time (Iterations)	42.25	46.9	50.55
	Average reward obtained per iteration	0.56	0.489	0.425
2 × Pursuers' motion speed = Evaders' motion speed	Average capturing time (Iterations)	48.05	58.7	72.6
	Average reward obtained per iteration	0.466	0.406	0.276

Table 2. Simulation results

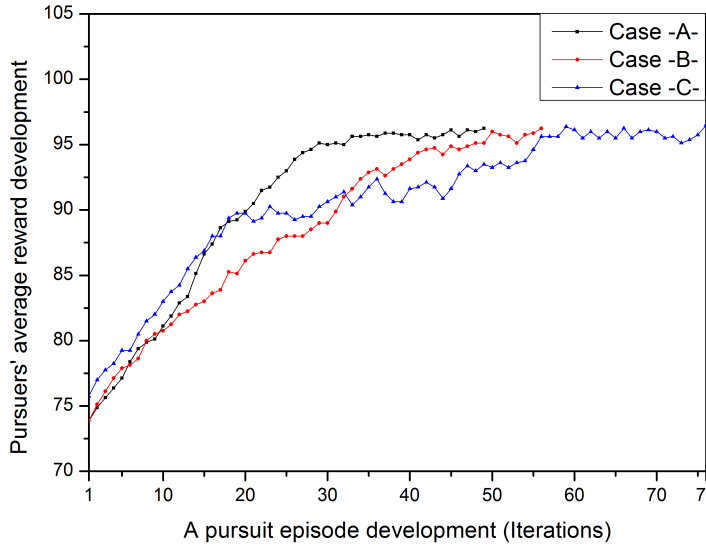


Figure 7. Pursuer's payoffs development during a complete pursuit ($2 \times$ pursuers' motion speed = evaders' motion speed)

namic reattribution of the roles (leader and followers). From this last study, we have concluded that the reorganization degree of the pursuit groups increases in relation to pursuers velocity. However, it is proportionally inverse to the capturing time.

6 CONCLUSIONS

In this paper, we proposed a new multi-agent path planning based reinforcement learning and leader-follower principles. The main objective of this work is to increase the collaboration level between the agents during the tasks' execution. In addition, this approach includes a certain dynamism regarding the roles of Leader and Follower according to the environment changes. Knowing that the principle of the followers is to follow the leader's path, the proposed path could be also used in a partially observable environment. In other words, the followers move to capture the evaders without knowing their exact positions. To reflect the feasibility of this work, we applied it to the PE game in comparison with recent path planning approaches. The simulation results proves that the new path planning improves the pursuit capturing time and also pursuers' payoff development during the pursuit. Furthermore, we have processed the PEG in the case where the evaders' velocity is superior to that

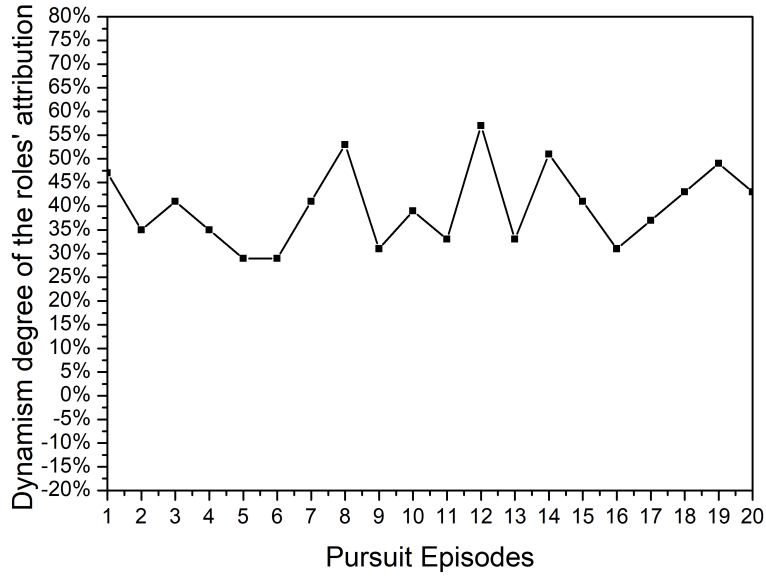


Figure 8. The study of the dynamism degree of the roles' attribution during 20 pursuit episodes (2 × pursuers' motion speed = evaders' motion speed)

of the pursuers. From this study, we have concluded that the dynamism of the pursuit groups increases in relation to the increase of the evaders' velocity. Regarding the pursuit capturing time, we can constate that the new proposed approach is less affected by the increase of the evaders' velocity in comparison with the other approaches.

On the other hand, we can constate that the proposed approach does not take into consideration the complex obstacles' processing. Therefore, in the future work, we will equip this path planning algorithm with a new obstacle avoidance method that takes into account the processing of complex and dynamic obstacles.

Regarding the real-world application, the proposed path planning algorithm can easily be applied to control the cooperative behavior of mobile robots or UAVs during the processing of complex tasks, such as warehouse automation or targets' capture. However, in case of UAVs or robots, these lasts must be equipped with a communicating system allowing them to calculate the distance between each other. Moreover, these entities must have access to some information about a part of each other (agents belonging to the same group), such as, their environment positions as well as their velocities.

REFERENCES

[1] Kuo, J. Y.—Yu, H. F.—Liu, K. F. R.—Lee, F. W.: Multiagent Cooperative Learning Strategies for Pursuit-Evasion Games. Mathematical Problems in Engineering, Vol. 2015, 2015, Art. No. 964871, doi: 10.1155/2015/964871.

[2] Durfee, E. H.: Scaling Up Agent Coordination Strategies. Computer, Vol. 34, 2001, No. 7, pp. 39–46, doi: 10.1109/2.933502.

[3] Lihui, Z.: On the Techniques of Multi-Agent Path Planning and the Collaboration. 2010 International Conference on Computer Application and System Modeling (ICCASM 2010), IEEE, Vol. 13, 2010, pp. 430–433, doi: 10.1109/ICCASM.2010.5622835.

[4] Horling, B.—Lesser, V.: A Survey of Multi-Agent Organizational Paradigms. The Knowledge Engineering Review, Vol. 19, 2004, No. 4, pp. 281–316, doi: 10.1017/S0269888905000317.

[5] Ferber, J.—Gutknecht, O.—Michel, F.: From Agents to Organizations: An Organizational View of Multi-Agent Systems. In: Giorgini, P., Müller, J. P., Odell, J. (Eds.): Agent-Oriented Software Engineering IV (AOSE 2003). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 2935, 2003, pp. 214–230, doi: 10.1007/978-3-540-24620-6_15.

[6] Dignum, V.—Aldewereld, H.—Dignum, F.: On the Engineering of Multi Agent Organizations. Proceedings of the 12th International Workshop on Agent-Oriented Software Engineering, 2011, pp. 53–65.

[7] El Habib Souidi, M.—Siam, A.—Pei, Z.—Piao, S.: Multi-Agent Pursuit-Evasion Game Based on Organizational Architecture. Journal of Computing and Information Technology, Vol. 27, 2019, No. 1, pp. 1–11, doi: 10.20532/cit.2019.1004318.

[8] Savall, M.—Pécuchet, J. P.—Chaignaud, N.—Itmi, M.: YAMAM – Un Modèle d'Organisation pour les Systèmes Multi-Agents. Implémentation dans la Plate-Forme Phœnix. Proceedings of 3ème Conférence Francophone de MOdélisation et SIMulation (MOSIM '01), 2001, pp. 25–27 (in French).

[9] Khalil, K. M.—Abdel-Aziz, M.—Nazmy, T. T.—Salem, A. B. M.: Machine Learning Algorithms for Multi-Agent Systems. Proceedings of the International Conference on Intelligent Information Processing, Security and Advanced Communication (IPAC '15), 2015, doi: 10.1145/2816839.2816925.

[10] Pei, Z.—Piao, S.—El Habib Souidi, M.—Qadir, M. Z.—Li, G.: Coalition Formation for Multi-Agent Pursuit Based on Neural Network. Journal of Intelligent and Robotic Systems, Vol. 95, 2019, No. 3, pp. 887–899, doi: 10.1007/s10846-018-0893-6.

[11] Abiodun, O. I.—Jantan, A.—Omolara, A. E.—Dada, K. V.—Mohamed, N. A.—Arshad, H.: State-of-the-Art in Artificial Neural Network Applications: A Survey. Heliyon, Vol. 4, 2018, No. 11, Art. No. e00938, doi: 10.1016/j.heliyon.2018.e00938.

[12] 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.

[13] Lin, P.—Wang, Y.—Qi, H.—Hong, Y.: Distributed Consensus-Based K-Means Algorithm in Switching Multi-Agent Networks. Journal of Systems Science and Complexity, Vol. 31, 2018, No. 5, pp. 1128–1145, doi: 10.1007/s11424-018-7102-3.

[14] Avrachenkov, K.—Cottatellucci, L.—Maggi, L.: Cooperative Markov Decision Processes: Time Consistency, Greedy Players Satisfaction, and Cooperation Maintenance. International Journal of Game Theory, Vol. 42, 2013, No. 1, pp. 239–262, doi: 10.1007/s00182-012-0343-9.

[15] Buşoniu, L.—Babuška, R.—De Schutter, B.: Multi-Agent Reinforcement Learning: An Overview. In: Srinivasan, D., Jain, L. C. (Eds.): Innovations in Multi-Agent Systems and Applications - 1. Springer, Berlin, Heidelberg, Studies in Computational Intelligence, Vol. 310, 2010, pp. 183–221, doi: 10.1007/978-3-642-14435-6_7.

[16] de Lima, J. V. C. F.—Belo, E. M.—da Silva Marques, V. A.: Multi-Agent Path Planning with Nonlinear Restrictions. Evolutionary Intelligence, Vol. 14, 2021, No. 1, pp. 191–201, doi: 10.1007/s12065-020-00534-1.

[17] Rong, T. X.—Mei, X. H.: Robot Path Planning Based on A* Algorithm and Genetic Algorithm. 2020 International Conference on Computer Engineering and Intelligent Control (ICCEIC), IEEE, 2020, pp. 101–105, doi: 10.1109/ICCEIC51584.2020.00028.

[18] Abhishek, B.—Ranjit, S.—Shankar, T.—Eappen, G.—Sivasankar, P.—Rajesh, A.: Hybrid PSO-HSA and PSO-GA Algorithm for 3D Path Planning in Autonomous UAVs. SN Applied Sciences, Vol. 2, 2020, No. 11, Art. No. 1805, doi: 10.1007/s42452-020-03498-0.

[19] Biswas, S.—Anavatti, S. G.—Garratt, M. A.: Obstacle Avoidance for Multi-Agent Path Planning Based on Vectorized Particle Swarm Optimization. In: Leu, G., Singh, H. K., Elsayed, S. (Eds.): Intelligent and Evolutionary Systems. Springer, Cham, Proceedings in Adaptation, Learning and Optimization, Vol. 8, 2017, pp. 61–74, doi: 10.1007/978-3-319-49049-6_5.

[20] Zheng, Y.—Li, B.—An, D.—Li, N.: A Multi-Agent Path Planning Algorithm Based on Hierarchical Reinforcement Learning and Artificial Potential Field. 2015 11th International Conference on Natural Computation (ICNC), 2015, pp. 363–369.

[21] Asil, Ü.—Uslu, E.: Potential Field Path Planning and Potential Field Path Tracking. 2020 Innovations in Intelligent Systems and Applications Conference (ASYU), IEEE, 2020, pp. 1–4, doi: 10.1109/ASYU50717.2020.9259808.

[22] Zhu, Z. Y.—Liu, C. L.: A Novel Method Combining Leader-Following Control and Reinforcement Learning for Pursuit Evasion Games of Multi-Agent Systems. 2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV), IEEE, 2020, pp. 166–171, doi: 10.1109/ICARCV50220.2020.9305441.

[23] Xu, L.—Hu, B.—Guan, Z.—Cheng, X.—Li, T.—Xiao, J.: Multi-Agent Deep Reinforcement Learning for Pursuit-Evasion Game Scalability. In: Jia, Y., Du, J., Zhang, W. (Eds.): Proceedings of 2019 Chinese Intelligent Systems Conference (CISC 2019). Springer, Singapore, Lecture Notes in Electrical Engineering, Vol. 592, 2019, pp. 658–669, doi: 10.1007/978-981-32-9682-4_69.

[24] Hwang, K. S.—Chen, Y. J.—Jiang, W. C.—Lin, T. F.: Continuous Action Generation of Q-Learning in Multi-Agent Cooperation. Asian Journal of Control, Vol. 15,

2013, No. 4, pp. 1011–1020, doi: 10.1002/asjc.614.

[25] Bilgin, A. T.—Kadioglu-Urtis, E.: An Approach to Multi-Agent Pursuit Evasion Games Using Reinforcement Learning. 2015 International Conference on Advanced Robotics (ICAR), IEEE, 2015, pp. 164–169, doi: 10.1109/ICAR.2015.7251450.

[26] Zheng, Y.—Fan, W.—Han, M.: Research on Multi-Agent Collaborative Hunting Algorithm Based on Game Theory and Q-Learning for a Single Escaper. Journal of Intelligent and Fuzzy Systems, Vol. 40, 2021, No. 1, pp. 205–219, doi: 10.3233/JIFS-191222.

[27] Ehsan, E.—Kunwar, F.: Probabilistic Search and Pursuit Evasion on a Graph. Transactions on Machine Learning and Artificial Intelligence, Vol. 3, 2015, No. 3, pp. 56–65, doi: 10.14738/tmlai.33.1313.

[28] Souidi, M. E. H.—Piao, S.—Li, G.: Mobile Agents Path Planning Based on an Extension of Bug-Algorithms and Applied to the Pursuit-Evasion Game. Web Intelligence, Vol. 15, 2017, No. 4, pp. 325–334, doi: 10.3233/WEB-170369.

[29] Sezer, V.—Gokasan, M.: A Novel Obstacle Avoidance Algorithm: "Follow the Gap Method". Robotics and Autonomous Systems, Vol. 60, 2012, No. 9, pp. 1123–1134, doi: 10.1016/j.robot.2012.05.021.

[30] Basak, S.—Mazumdar, B. D.: Multi-Agent Coalition Formation for Course Selection Strategies in E-Learning System. International Journal of Mathematical Trends and Technology (IJMTT), Vol. 6, 2014, No. 1, pp. 36–43, doi: 10.14445/22315373/IJMTT-V6P503.

[31] Tisue, S.—Wilensky, U.: NetLogo: A Simple Environment for Modeling Complexity. Proceedings of the Fifth International Conference on Complex Systems (ICCS 2004), 2004, pp. 16–21.

Mohammed El Habib Souidi is Lecturer in the Department of Mathematics and Computer Science, Faculty of Sciences and Technology, University of Khenchela, Algeria. He received his Ph.D. in computer science from the Harbin Institute of Technology (China), in 2017. His main areas of research include multiagent task coordination, reinforcement learning, game theory and path planning.

Makhlouf Ledmi received his Ph.D. in computer science from University of Batna 2, Algeria, in 2020. He is currently Associate Professor at the Mathematics and Computer Science Department at University of Khenchela and member of Knowledge Management Group ICOSI Lab. His research interests include data mining, soft computing, and bioinformatics.

Toufik Messaoud Maarouk is Lecturer in the Department of Mathematics and Computer Science, Faculty of Sciences and Technology, University of Khenchela, Algeria. He received his Ph.D. in computer science from the Constantine University, Algeria, in 2012. His main areas of research include formal methods, concurrency theory, formal semantics and distributed computing.

Abderrahim Siam received his Ph.D. in computer science from the University of Constantine, Algeria. He is working as Professor in the Department of Mathematics and Computer Science in University of Khenchela, Algeria. He is currently the Vice Rector of University of Khenchela, Algeria. His research interests include software engineering, fuzzy logics, formal methods, multiagent systems and complex systems.

Abdeldjalil Ledmi received his Engineer's degree in computer science from the University of Khenchela, Algeria in 2010. He also received his Master's degree in computer science from the same university in 2014, and his Ph.D. in computer science from the Larbi Tebessi University, Algeria in 2019. He is working as Lecturer in the Department of Mathematics and Computer Science at the University of Khenchela, Algeria. Moreover, he is affiliated as a Researcher in ICOSI Lab.

Computing and Informatics, Vol. 42, 2023, 1184–1212, doi: 10.31577/cai_2023_5_1184

METAHEURISTIC FOR SOLVING THE DELIVERY MAN PROBLEM WITH DRONE

Ha-Bang BAN*, Hai-Dang PHAM

School of Information and Communication Technology
Hanoi University of Science and Technology
e-mail: BangBH@soict.hust.edu.vn, HaiPD@soict.hust.edu.vn

Abstract. Delivery Man Problem with Drone (DMPD) is a variant of Delivery Man Problem (DMP). The objective of DMP is to minimize the sum of customers' waiting times. In DMP, there is only a truck to deliver materials to customers while the delivery is completed by collaboration between truck and drone in DMPD. Using a drone is useful when a truck cannot reach some customers in particular circumstances such as narrow roads or natural disasters. For NP-hard problems, metaheuristic is a natural approach to solve medium to large-sized instances. In this paper, a metaheuristic algorithm is proposed. Initially, a solution without drone is created. Then, it is an input of split procedure to convert DMP-solution into DMPD-solution. After that, it is improved by the combination of Variable Neighborhood Search (VNS) and Tabu Search (TS). To explore a new solution space, diversification is applied. The proposed algorithm balances diversification and intensification to prevent the search from local optima. The experimental simulations show that the proposed algorithm reaches good solutions fast, even for large instances.

Keywords: DMPD, metaheuristic, VNS

1 INTRODUCTION

Delivery Man Problem with Drone (DMPD) is a variant of Delivery Man Problem (DMP) that is seen as a "customer-centric" routing problem. However, in DMP, there is only a truck to deliver materials to customers. In the case of narrow or

* Corresponding author

congested roads and disasters, when a truck cannot reach some customers, the drone is used. Informally, in DMPD, there is a truck and drone at the main depot s, and n customers. The goal is to find a tour with a combination of truck and drone that minimizes the overall customers' waiting times while ensuring that all customers are served.

Using drone to deliver parcels to customers is not new in the literature. Many companies use drones for parcel delivery [1] because drone brings many advantages:

1. It can be operated on itself;
2. It is beneficial in the case of congestion;
3. It can move faster than trucks [2].

However, a drone also has some drawbacks:

1. The drone can carry parcels of 2 kilograms. It cannot carry packages that exceed its weight [3]. Therefore, it needs to return to the depot after each delivery;
2. Its range is limited because of the battery-powered engines. On the other hand, the truck has a long range and brings many materials but is rather heavy and slow.

Therefore, the collaboration between trucks and drones brings many advantages. The idea of the collaboration is that the delivery truck and drone collaboratively serve all customers. Figure 1 describes an example in which 8 customers need to be visited. We see that the drone visits two customers instead of the truck. The truck is not waiting and can serve another customer instead. The truck or drone that first reaches the reconnection node has to wait for each other. By the parallelization of delivery tasks, the total waiting time can be reduced.

To the best of our knowledge, though various works were proposed to solve DMP and its variants [4, 5, 6, 7, 8, 9, 10, 11, 12, 13], there is no work for DMPD in the literature. Due to the characteristics of the problem, adapting these metaheuristics for DMP to solve DMPD is not easy. We aim to develop an effective metaheuristic to solve the problem. The success of any metaheuristic approach depends on the right balance between intensification and diversification. The main contributions of this work can be summarized as follows:

- From the algorithmic perspective, the proposed metaheuristic consists of five steps. We introduce a split heuristic that builds a DMPD solution from a DMP solution. In the local search step, the VNS with new neighborhoods adapted from the traditional ones is proposed for DMPD. This step aims to exploit the explored solution space. Lastly, a diversification step is applied to lead our search to a new region. Moreover, Tabu Search is incorporated into our algorithm to prevent the search from local optima.
- From the computational perspective, extensive numerical experiments on benchmark instances show that our algorithm reaches good solutions fast, even for

large instances. In addition, the proposed algorithm is adapted to solve DMPD. The outcome shows that the proposed algorithm obtains good solutions fast. Its solutions are compared with the state-of-the-art metaheuristics' ones.

The rest of this paper is organized as follows: Sections 2, and 3 present the literature and the problem definition, respectively. Section 4 describes the proposed algorithm. Computational evaluations are reported in Section 5. Sections 6 and 7 discuss and conclude the paper, respectively.

2 LITERATURE

DMPD is NP-hard because it is a generalization case of DMP. Despite the similarities between DMP and DMPD, solving DMPD is more challenging due to drone's presence. Currently, there are three main methods to solve an NP-hard problem, specifically:

1. exact algorithms,
2. approximation algorithms, and
3. heuristic (or metaheuristic) algorithms.

The exact algorithms obtain the optimal solution, but they consume more time. Therefore, it can only solve the problems with small sizes. Meanwhile, for an α-approximation algorithm, the value of α is often large, and they are complex to implement. Metaheuristic, on the other hand, is naturally suitable to solve large-sized instances in a short time.

2.1 DMP

Though we found no work for DMPD, several classes of problems closely related to the problem were introduced in the literature: the Delivery Man Problem (DMP), DMP with Profits (DMPP), and DMP in post-disaster.

- Delivery Man Problem (DMP) is a particular case where DMPD has no drone. Numerous works for DMP can be found in [6, 7, 8, 9, 10, 11, 13]. Ban et al. [7] also present an exact algorithm to solve instances with up to 40 vertices. Several approximation algorithms [10, 11, 13] are proposed to solve DMP with the best ratio of 3.59. In the metaheuristic algorithms, these algorithms [6, 8, 9] obtain good solutions fast for instances with up to 1000 vertices.
- DMP with Profits (DMPP) aims to find a travel plan for a server that maximizes the total revenue. However, contrary to DMP, in DMPP, not all the customers need to be visited. Metaheuristic algorithms in [5, 14] can solve well the problem with up to 500 vertices. After that, Dewilde et al. [12] have introduced a stochastic variant of DMPP under uncertain travel times. The difference between DMPP and stochastic DMPP is that the travel times in stochastic DMPP

are uncertain. The stochastic DMPP is heuristically solved using a beam search heuristic.

- In the minimizing latency in the post-disaster road clearance operations (ML-RCP) [4], we find a tour to minimize latency sum in the post-disaster. Their metaheuristic algorithm reaches the optimal or near-optimal solutions on Istanbul data within seconds.

These algorithms are the state-of-the-art algorithms for DMP. However, there is no drone presence, and they cannot be adapted directly to DMPD. That means that we cannot use the above algorithms to solve DMPD.

2.2 Routing with Drone

The UAV for surveillance purposes has been used for a long time. However, in recent times, research for drone delivery has been improved by logistics companies. We describe the works related to routing with drone.

- Murray et al. [15] first introduced the TSP with drone (TSPD) in which there are a single truck and a single drone. In this paper, two constraints are considered. In the first constraint, the drone can deliver only one parcel at a time because of the capacity limit. The second one shows that a subset of the customers can be visited by a drone. They proposed a mixed integer programming formulation and a heuristic to minimize the completion time for all vehicles.
- Agatz et al. [16, 17] proposed TSPD with different constraints in comparison with Murray et al. in which the drone is allowed to launch and return to the same location (the constraint is forbidden in [15]). They developed a new dynamic programming approach to solve exactly the problem with up to 10 vertices.
- Freitas and Penna [18] proposed a metaheuristic with the min-time objective function. Firstly, a mixed-integer program (MIP) is used to obtain an initial solution. Then it is modified by removing some truck vertices and adding some drone ones. Finally, several neighborhoods are applied to find better solutions. The experiment shows that the delivery time is decreased up to 68 %.
- Poikoene et al. [19] proposed a branch-and-bound to solve TSPD in the case of allowing drone to revisit customers. They tested the proposed algorithm on small instances. The experimental results indicate that the problem can be solved optimally for the problem with 10 customers.
- Poikoene et al. [1] have extended TSPD with more constraints. Specifically, a drone can serve multiple customers in a row. Moreover, the capacity limit of a drone is also mentioned. They proposed a flexible heuristic that obtains good solutions fast. It showed that metaheuristic is a suitable approach for DMPD.
- Ha et al. [20] proposed a metaheuristic algorithm to minimize the total operational cost (the min-cost TSPD). They proposed the greedy randomized adaptive search procedure (GRASP) to minimize the total operational cost (the min-cost

TSPD) [20]. The results are quite good in terms of solution quality and running time. They then presented a hybrid genetic algorithm (GA) [21] with some new strategies to control diversity. The results are provided for instances with up to 50 and 100 customers.

- TSPD with more than one drone is also interested by many researchers in literature [22, 23, 24, 25, 26, 27].

The above works are the state-of-the-art algorithms to solve TSPD. DMPD in this paper is different from TSPD. Our objective function is to minimize the total waiting time of customers (customer-oriented routing), while their works aim to minimize the minimum travel cost (server-oriented routing). The difference between the two objective functions leads to a big difference in their solutions. In [28], Salehipour et al. showed that the good algorithms for the TSP cannot be adapted for DMP. Therefore, developing an efficient algorithm for DMPD is necessary.

3 PROBLEM DEFINITION

We are given a complete graph $K_n = (V, E)$ with $V = V' \cup \{v_1\}$ where v_1 is a depot and $V' = V^t \cup V^d \cup V^c$ in which V^t, V^d, V^c are the set of vertices visited by only truck, only drone, and both of them, respectively. Let $C = \{c_{ij} \mid i, j = 1, \ldots, n\}$ be the distance matrix between all vertices. The travel time between a pair of vertices (v_i, v_j) is the distance between v_i and v_j (c_{ij}). Let $\rho = \frac{\beta}{\alpha}$ be the ratio between the drone's and truck's travel time per unit distance. Every customer can be served by either a truck or drone. A drone can only deliver one parcel at a time. We make the following assumptions:

- A truck can dispatch and pick up a drone only at the depot or at a customer location.
- Customers can only be served once, either by a drone or a truck.
- DMPD allows the parallelization of delivery tasks. That means the drone departs from the truck at the customer i. It then drops off a parcel at the second customer j and goes to the connection customer k. The drone cannot serve more than one customer between nodes i and j.
- The vehicle (truck or drone) that first arrives at the reconnection node has to wait for the other one.
- When returning to the truck to take another parcel, the time required to prepare the drone launch is negligible.

Our objective is to minimize the total waiting time of all customers.

Because of the presence of the truck and the drone, a DMPD solution $T = (TR, DR)$ is represented by two components:

- A truck tour, denoted as TR, is a sequence of vertices: $TR = \{v_1, v_2, \ldots, v_k, \ldots\}$.

- A drone tour, denoted as DR, is a sequence of tuple $< i, j, k >$. That means drone launches from v_i, delivers to v_j, and rejoins truck at v_k.

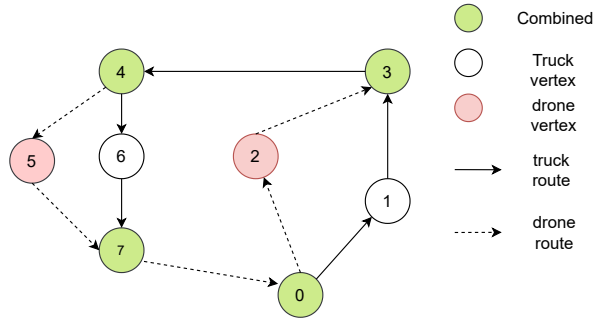


Figure 1. The simple example of DMPD

After that, we describe a simple example of DMPD. There are three types of vertices. A truck vertex is a vertex that is delivered by truck. Similarly, a drone vertex is a vertex delivered by a drone. A combined vertex is a vertex visited by both the truck and drone. For example, there are 7 vertices in the graph in which vertex 1 and 6 are truck vertices, while vertex 2 and 5 are drone vertices. Last, vertex 3, 4, and 7 are combined vertices. The truck moves from the depot to vertex 1 while the drone is flown from the depot to vertex 2. After that, truck travels from vertex 1 to 3, and drone moves from vertex 2 to 3. The truck and drone are rejoined at vertex 3. The truck has to wait for the drone if it goes to vertex 3 before the drone. On the other hand, the drone has to wait for the truck if it goes to vertex 3 before the truck. The truck and drone travel together to vertex 4. The drone is launched toward vertex 5 while the truck goes to vertices 4 to 6. The truck and drone are rejoined at vertex 7. The waiting time of vertices is calculated as follows:

$$
\begin{aligned}
w(v_0) &= 0, \\
w(v_1) &= \alpha \times c(v_0, v_1), \\
w(v_2) &= \beta \times c(v_0, v_2), \\
w(v_3) &= w(v_1) + \alpha \times c(v_1, v_3), \\
w(v_4) &= \max\{w(v_1) + \alpha \times c(v_1, v_3), w(v_2) + \beta \times c(v_2, v_3)\} + \alpha \times c(v_3, v_4), \\
w(v_5) &= w(v_4) + \beta \times c(v_4, v_5), \\
w(v_6) &= w(v_4) + \alpha \times c(v_4, v_6),
\end{aligned}
$$

$$w(v_7) = w(v_6) + \alpha \times c(v_6, v_7),$$

$$W(T) = \sum_{i=0}^{7} w_i.$$

If the drone first arrives at v_3, v_7, it has to wait for the truck because the drone cannot serve more than one customer between nodes i and j. It means that customers at v_3, v_7 must be served by the truck. The waiting time of customer at v_4 is calculated more complex than the others because of rendezvous time.

The objective function in this example is different from the one of TSPD when it optimizes the following function cost: $\max\{\alpha \times (c(v_0, v_1) + c(v_1, v_3)), \beta \times (c(v_0, v_2) + c(v_2, v_3))\} + \alpha \times c(v_3, v_4) + \max\{\alpha \times (c(v_4, v_6) + c(v_6, v_7)), \beta \times (c(v_4, v_5) + c(v_5, v_7))\} + \alpha \times c(v_7, v_0)$.

4 A METAHEURISTIC ALGORITHM

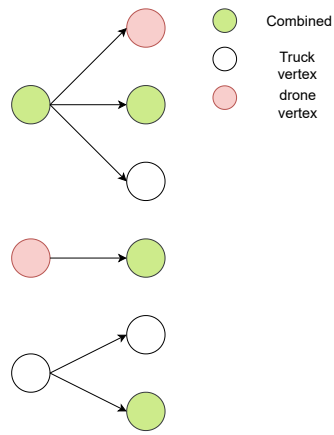


Figure 2. Label assignment for a vertex

The algorithm is divided into five steps: In the first step, a solution without the drone is created. Then, it is an input of splitting procedure to convert DMP-solution into DMPD-solution in the second. After that, it is improved by the combination of Variable Neighborhood Search (VNS) and Tabu Search (TS) in the local search. When the proposed algorithm gets stuck into local optima, diversification is used. It leads the search to unexplored solution space. Algorithm 1 depicts the whole process. In line 2, we build a giant DMP tour T_{TR} visiting the depot and all the customers. For that aim, we apply some heuristics for construction. As a result, an initial solution T_{TR}, where all vertices are visited by truck and no vertex is assigned to drone. Line 5 decomposes T_{TR} in two partitions: the first assigned to

Algorithm 1 General scheme of algorithm

Input: $v_1, V, N_i(T)(i = 1, \ldots, 6), level$ are a starting vertex, the set of vertices in K_n, the set of neighborhoods, the parameter to control the strength of the perturbation procedure, respectively.

Output: the best solution T^*.

{Step 1: Construction-DMP}
$T_{TR} \leftarrow$ construction-DMP$(v_1, V, N_i(T), K_n)$; {T_{TR} is DMP tour}
{Step 2: Split phase}
$T = (T_t, T_d) \leftarrow$ split(T_{TR}); {Assign vertices into "drone", "truck", or "combined" label}
$T^* \leftarrow T$;
while The stop condition is not satisfied **do**
  $T^{'} \leftarrow T$;
  {Step 3: Local search}
  $k = 1$;
  repeat
    Find the best neighborhood $T^{''}$ of $T \in N_k(T^{'})$;
    if $((W(T^{''}) < W(T^{'})$ and ($T^{'}$ is not Tabu)) $\|$ $(W(T^{''}) < W(T^*))$ **then**
      $T^{'} = T^{''}$;
      Update Tabu list;
      if $((W(T^{''}) < W(T^*))$ **then**
        $T^* = T^{''}$;{Update the best solution}
      $k = 1$;
    else
      $k = k + 1$; {switch to another neighborhood}
  until $k < k_{max}$;
  {Step 4: Intensification}
  if $W(T^{''}) < W(T)$ **then**
    $\overline{T}$ = Perform VNS without using tabu on the solution $T^{''}$;
    if $(W(T^{''}) < W(\overline{T}))$ or $(W(T^*) < W(\overline{T}))$ **then**
      $T^{''} \leftarrow \overline{T}$;
      if $(W(T^*) < W(\overline{T}))$ **then**
        $T^* \leftarrow \overline{T}$; {Update the best solution}
    $T \leftarrow T^{''}$;
  {Step 5: Diversification}
  $T \longleftarrow$ Perturbation$(T, level)$;
return T^*;

Algorithm 2 GRASP + VNS

Input: v_1, V, γ are a starting vertex, the set of vertices in K_n, and the size of RCL, respectively.
Output: the initial solution T.

```
T ← v_1;
while |x| < n do
    Create RCL with γ vertices v_i ∈ V closest to v_l; // v_l is the last vertex in T;
    Select randomly vertex v = {v_i | v_i ∈ RCL and v_i ∉ T};
    T ← T ∈ {v};
repeat
    T' ← Shaking(T);
    T'' ← arg min N_k(T'); {local search}
    if (L(T'' < L(T)) then
        T ← T'' {update solution}
    else
        k = k + 1; {switch to another neighborhood}
until k = k_max
return T;
```

truck (T_t) and the second assigned to drone (T_d). Lines 9–22 optimize the tour T by using six neighborhoods in local search. In lines 23–33, the VNS step is implemented without Tabu in the intensification step. Line 15 implements Perturbation to maintain diversity. The algorithm is repeated until the stop condition is satisfied.

4.1 Construction-DMP

In the first step, we generate a DMP solution from some heuristics. In this paper, three heuristics and an exact algorithm are used as follows:

- k-nearest neighbor [29]: It is inspired by the well-known nearest-neighbor algorithm. It begins from the root and repeatedly travels each vertex that is chosen among k closest vertices randomly.
- Insertion heuristic [29]: It works similarly to the above heuristic, but it picks a random vertex among all unvisited nodes iteratively.
- GRASP with VNS: The version of GRASP construction phase [30] is used to create an initial solution, and then it is improved by the VNS [31, 30]. Initially, at each vertex, a Restricted Candidate List including its nearest vertices is built. At a constructive step, a vertex is selected randomly from RCL, and added into the solution. The GRASP stops when all vertices are visited. After that, the GRASP's solution is shaken by swapping vertices randomly before the VNS step is used to improve the solution. The shaking aims to maintain the diversity of our search. The VNS is based on a simple principle that systematically switches between different neighborhoods. In the VNS, we use some popular

Algorithm 3 Greedy approach for splitting

Input: v_1, V are a starting vertex and graph, respectively.
Output: the best solution T^*.

$v_1.label$ = "combined";
$TR = TR \cup \{v_1\}$;
$i = 0$;
while $\exists v$ is unvisited **do**
  $v = T[i]$;
  if ($v.label ==$ "truck") **then**
    rd =random(2);
    if (rd ==1) **then**
      $v'.label$ = "truck";
      $TR = TR \cup \{v^{'}\}$;
    else
      $v.label$ = "drone";
      $TD = TD \cup \{v_1\}$;
  else if ($v.label ==$ "drone") **then**
    for $k \leftarrow i+1$ to n **do**
      find a $v^{'} = \{v_j | v_j$ is unvisited and $T(i-1,j,k) = \min_{j=i+1}^{k-1} T(i,j,k)\}$;
    $v.label$ = "combined";
  else
    rd =random(2);
    if (rd ==1) **then**
      $v.label$ = "truck";
    else
      $v.label$ = "drone";
  $T = T \cup \{v_1\}$;
return T^*;

neighborhoods such as remove-insert (N_1), swap (N_2), and 2-opt (N_3) [30]. The combination between the GRASP and VNS demonstrated the efficiency in [8]. The GRASP + VNS scheme is described in Algorithm 2.

- Exact algorithm: We can use an exact algorithm in [7] to obtain the optimal solution for DMP. However, the exact algorithm can solve the problem with small sizes (less than 40 vertices).

4.2 Split Scheme

In this step, two approaches are used: Greedy heuristic and exact partitioning algorithm based on dynamic programming. All approaches remain in the sequence of vertices visited.

Algorithm 4 Dynamic-Programming for splitting(K_n, C)

Input: K_n, C are the graph, and the distance matrix, respectively.
Output: A matrix M, a list of drone vertices DM.

for $i \leftarrow 1$ to n **do**
  for $j \leftarrow 1$ to n **do**
    $M[i,j] \leftarrow 0$;
    $DM[i,j] \leftarrow 0$;
for $i \leftarrow 1$ to n **do**
  for $j \leftarrow 1$ to n **do**
    $TT \leftarrow -Inf$;
    for $k \leftarrow i$ to j **do**
      $TTd = \beta \times (c(v_i, v_k) + c(v_k, v_j))$.
      $TTt = \alpha \times (\sum_{h=i}^{k-2} c(v_h, v_{h+1}) + \sum_{h=k+1}^{j-1} c(v_h, v_{h+1}))$.
      if $(TT \leq \max\{TTd, TTt\})$ **then**
        $TT = \max\{TTd, TTt\})$
        $DM[i,j] = k$; {drone position is stored}
    $M[i,j] = TT$;
$MT[1] \leftarrow 0$;
$DM = \phi$;
for each v_i in V **do**
  w=Inf;
  for $k \leftarrow 1$ to i **do**
    if $(MT[k] + M[k,i] < w)$ **then**
      $w = MT[k] + M[k,i]$;
      $DP = k$;
  $DM = DM \cup DP$; {DM is a list of drone positions}
  $MT[i] = w$;
  Assign a vertex at DP position as "drone" label.
return DM;

4.2.1 Greedy Approach

We propose a greedy heuristic to split an initial T_{TR} into two subtours:

1. a subtour for the drone;
2. and a subtour for the truck.

Each vertex $v \in V$ is assigned by a label. The "truck" label indicates the truck, while the "drone" label indicates the drone. Moreover, a vertex that both the truck and drone can travel is called "combined". Our greedy approach includes two phases. An initial solution consisting of the truck and drone is generated in the first phase, while "drone" label assignment optimization is done in the second. In this first stage, the label of all vertices is "unvisited". In every heuristic step, a vertex is assigned to the "truck" or "drone" label. The algorithm stops when there is no

Algorithm 5 Perturbation($T, level$)

Input: T, k are the tour, and the number of swap, respectively.
Output: a new tour T.

```
TR, TD = get(T);
while (level > 0) do
    {Perturbation for truck}
    i_TR = rand(TR);
    j_TR = rand(TR);
    TR=swap(i_TR, j_TR, TR);
    {Perturbation for drone}
    i_T D = rand(TD);
    j_T D = rand(TD);
    TD = swap(i_TD, j_TD, TD);
    level = level − 1;
T = (TR, TD);
return T;
```

"unvisited" vertex. The detail of the Split Algorithm is described in Algorithm 3. To assign a label for a current vertex v, we need to consider its previous one:

- If a previous vertex is "combined" one, there are three label candidates ("truck", "drone", and "combined") for the current vertex v (see Figure 2 a)). Which label brings the best total cost for the tour will be selected.
- If a previous vertex is the truck vertex, there are two options ("truck", and "combined") for the current vertex v (see in Figure 2 b)). We choose a label such that the tour receives the best cost.
- If a previous vertex is the drone vertex, the label of the current vertex is "combined" (see Figure 2 c)). This is the only option in this case.

After the first step, we optimize the "drone" label assignment in the second one. An operation is defined by a triplet (i, j, k), which starts at v_i, delivers at v_k by drone, and rejoins at v_j. We try to enumerate the different cases resulting from changing a set of v_k candidates. After that, a v_k is picked so that our solution reaches the best cost. For example, in Figure 3, the drone and truck can rejoin at v_3, v_4, and v_n. Vertex v_4 is chosen because rejoining at this vertex makes our solution's cost decrease. The time complexity of the greedy heuristic is $O(n^2)$.

4.2.2 Dynamic Approach

We now describe an algorithm to partition an initial solution $T = \{v_1, v_2, \ldots, v_k, \ldots, v_n\}$ into a truck tour TR and a drone tour TD based on dynamic programming. The order of vertices in T remains unchanged. For each vertex, the algorithm decides

to visit it by the truck or drone. A move is defined by a triplet (i, j, k), which starts at v_i, delivers at v_k by the drone, and rejoins at v_j. If a move does not contain the drone, we set k to -1. We also define $T(i, j, k)$ by the maximum amount of time to complete (i, j, k):

$$T(i,j,k) = \max\left\{\beta \times (c(v_i, v_k) + c(v_k, v_j)), \alpha \times \left(\sum_{h=i}^{k-2} c(v_h, v_{h+1})\right) + \sum_{h=k+1}^{j-1} c(v_h, v_{h+1}) + c(v_{k-1}, v_{k+1})\right\}.$$

For each subsequent move $(v_i, v_{i+1}, \ldots, v_j)$ in the tour, the minimum time is calculated as follows:

$$T(i,j) = \min_{k=i+1}^{j-1} T(i,j,k).$$

The recursive formulation for computing the minimum arrival time for the truck to reach a vertex is:

$$W(0) = 0,$$
$$W(v_i) = \min_{k=0}^{i-2}(W(v_k) + T(k, i-1) + \alpha \times c(v_{i-1}, v_i)).$$

The $\arg\min_{k=0}^{i-2}(W(v_k) + T(k, i-1) + \alpha \times c(v_{i-1}, v_i))$ is used to determine which of vertex is to be visited by the drone or truck.

4.3 Local Search

In this step, we apply the VNS [32, 33] for a full solution $T = (TR, TD)$. Neighbor solutions are evaluated, and the best feasible neighboring solution is accepted if it is non-tabu, improving, or tabu but globally improving. Six of our neighborhoods are inspired by traditional moves. We then describe these neighborhoods:

- Exchange role (N_1) exchanges the drone vertex to the truck vertex and vice versa. An example is shown in Figure 3.
- Swap (N_2) tries to swap the positions of each pair of vertices in T. An example is shown in Figure 4.
- Reverse-Swap (N_4) removes each pair of vertices from T and reverses the sub-tour between them. An example is shown in Figure 5.
- 2-opt (N_4) removes each pair of vertices from T and reverses the sub-tour between them. An example is shown in Figure 6.
- Add-drone (N_6) selects a truck location and replaces its delivery with drone delivery. An example is shown in Figure 7.

- Remove-drone (N_5) selects a drone location and replaces its delivery with a truck delivery. An example is shown in Figure 8.

After the local search step, its cost value is less than the current best value the new best value is updated. Moreover, all moves are updated in Tabu list.

4.4 Intensification and Diversification

When we find a good solution space, we try to exploit it in an intensification step. To exploit good solution space, we implement Step 2 without any tabu move. After that, the algorithm goes to the perturbation step to maintain diversification. A mechanism design plays an important role in obtaining success. If too small moves are applied, the search may get stuck into the previously visited solution space. On the other hand, excessive moves may drive the search to unexpected regions. In this work, we use two different perturbation methods for truck and drone. A perturbation method is simple to exchange some vertices visited by truck and drone for truck and drone, respectively. The detail of the step is described in Algorithm 5.

4.5 Tabu Lists

A tabu list for the moves is included in the proposed algorithm. The tabu status is assigned to an element for θ iterations, where θ is randomly selected. We describe the tabu list for each move as follows:

Exchange role: A position of exchanged vertices cannot be implemented by the same type of move if it is tabu.

Swap move: Two vertices swapped cannot be swapped again if they are tabu.

2-opt and reverse moves: The moves applied to two vertices cannot be applied again to the same positions.

Remove-drone: A "drone" vertex remains a "drone" label if it is tabu.

Add-drone: A "truck" vertex remains a "truck" label if it is tabu.

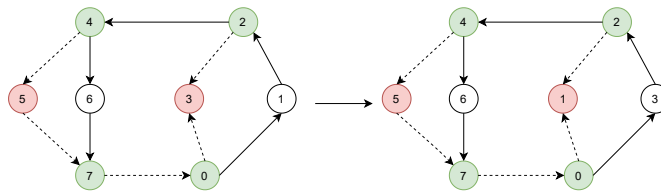


Figure 3. The exchange role

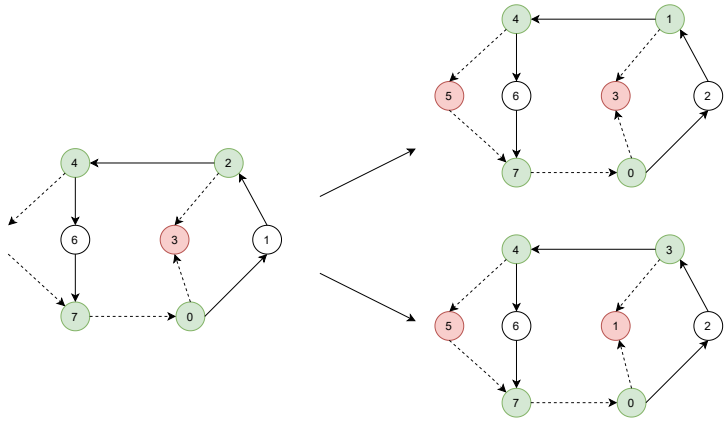


Figure 4. The swap

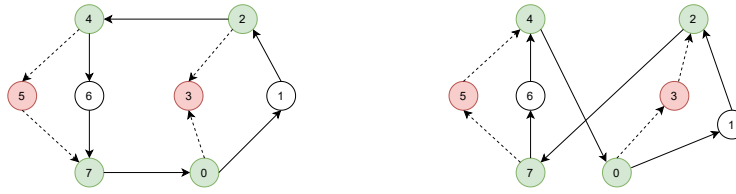


Figure 5. The reverse

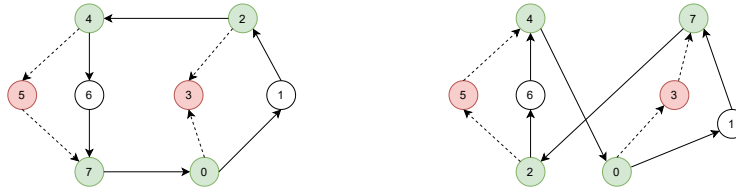
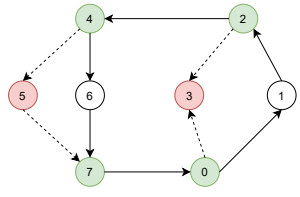


Figure 6. The 2-opt



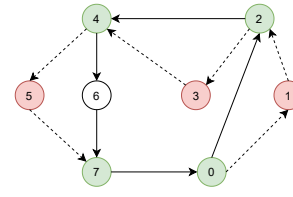


Figure 7. The drone insertion

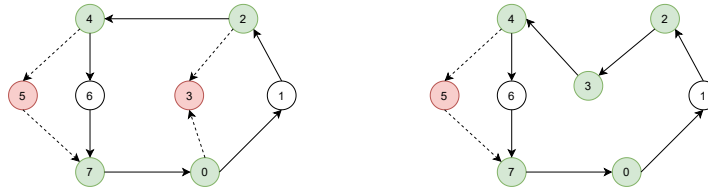


Figure 8. The drone removal

5 EVALUATIONS

Our algorithm is run on a Pentium 4 core i7 3.40 GHz 8 GB RAM processor. For all experiments, the parameters γ, *level*, and θ are set to 10, 5, and 5, respectively. These parameters are selected using empiric experiments and, with them, the algorithm might provide good solutions.

5.1 Instances

The proposed algorithm is tested on 900 instances in two datasets as follows:

- The first dataset is proposed by Poikonen et al. which are available on [19, 1]. This dataset includes 100 instances with up to 39 vertices. Customer locations in each instance are generated randomly from 50-by-50 grid. Moreover, the truck traveling time to move from customer i to j is calculated by $\lfloor |x_i - x_j| + |y_i - y_j| \rfloor$ while drone travel time is calculated by $\left\lfloor \frac{\sqrt{((x_i - x_j)^2 + (y_i - y_j)^2)}}{\rho} \right\rfloor$. The speed of drone is ρ times faster than truck speed. To evaluate the efficiency of the proposed algorithm with different speeds of truck and drone, our algorithm is tested with ρ values = 1, 2, and 3.
- The second dataset is released by Bouman [34]. This dataset consists of three types. In the first type (the uniform instances), the x and y coordinates are generated uniformly from $[0, 100]$. For the second type (the 1-center), an angle ϕ and distance r is created uniformly. The x and y coordinates are computed as $r \times \cos(\phi)$, and $r \times \sin(\phi)$. By doing this, locations close to the center $(0, 0)$ with higher probability than in the previous case. Finally, we have the 2-center, which is generated in the same way, but every location is transformed into 200 distance units over the x coordinate with probability 0.5. These instances have a greater probability of closing to two centers at $(0, 0)$ and $(200, 0)$. In all cases, the drone is ρ times as fast as the truck.

5.2 Results

The improvement of the proposed algorithm is defined considering *Best.Sol* (*Best.Sol* is the best solution found by our algorithm) in comparison with the other algorithms

as follows:

$$Gap_1[\%] = \left| \frac{Best.Sol - BKS}{BKS} \right| \times 100\,\%, \tag{1}$$

$$Gap_2[\%] = \left| \frac{Best.Sol - DMPS}{Best.Sol} \right| \times 100\,\%, \tag{2}$$

$$Improv[\%] = \left| \frac{Best.Sol - Init.Sol}{Init.Sol} \right| \times 100\,\%. \tag{3}$$

In all tables, *Init.Sol*, *Best.Sol*, *Aver.Sol*, T correspond to the initial, best, average solution, and average time in seconds of ten executions obtained by the proposed algorithm, respectively, while *BKS* is the best-known solution in the literature. DMPS corresponds to DMP solution in the first step. The experiment results can be found in Tables 1 to 18 in [35]. The values in Tables 1, 2 and 3 in this paper are the average values calculated from Tables 1 to 18 in [35].

Currently, no metaheuristic for this problem can be found in the literature to compare directly. Therefore, it is difficult to evaluate the efficiency of the proposed algorithm exactly. To overcome the issue, several state-of-the-art metaheuristic algorithms for TSPD are chosen to compare to our algorithm. Fortunately, Bouman et al. [36], and Murray et al. [15] support their code for solving TSPD. However, their algorithms are developed for TSPD rather than DMPD. We adapt their objective function to solve DMPD. Therefore, the proposed algorithm and their algorithms can be compared directly. Moreover, our solutions are compared to some algorithms [36, 15, 37] in the case of TSPD. All algorithms are run on the same instances.

Different experiments have been carried out to evaluate the efficiency of the proposed algorithms as well as analyze the impact of parameters: investigate the performance of different construction heuristics for DMP, explore the performance of different heuristics to convert from DMP to DMPD; compare DMPD solutions with solutions in the case of an only truck, analyze the impact of drone in developing the quality of solution, consider DMPD with some constraints, compare our solution quality with the previous algorithms in the case of TSPD.

5.3 The Impact of Different DMP's Construction Heuristics for DMPD

In this experiment, we evaluate the performance of different construction heuristics for DMP's solution. The experimental results can be found in Table 1.

Overall, all heuristics work with stable results. The GRASP with VNS provided the best solution quality in comparison with the other heuristics. Specifically, in Table 1, the average gap of k-nearest neighbor and Insertion heuristic are $-21.93\,\%$ and $-21.44\,\%$, respectively, while the average one of GRASP + VNS is $-25.71\,\%$.

It is understandable because GRASP with VNS balances diversification and intensification while the others only maintain intensification. In addition, we carried out some additional tests and found that, in general, using the optimal DMP tours does not obtain the better solution quality for DMPD while it consumes more time to solve. From these analyses, we decided to use GRASP with VNS to generate DMP tours in the next experiments.

α, β	Instance	NN		IH		GRASP	
		Gap_2	T	Gap_2	T	Gap_2	T
1, 1	Uniform	−12.77	2.09	−13.26	2.56	−18.58	2.83
	SingleCenter	−14.56	2.19	−13.44	2.53	−23.21	2.53
	DoubleCenter	−15.85	2.07	−15.37	2.61	−18.37	2.63
1, 2	Uniform	−22.09	2.12	−22.01	2.60	−23.29	2.64
	SingleCenter	−24.51	2.17	−22.56	2.53	−24.68	2.50
	DoubleCenter	−24.97	2.05	−23.98	2.71	−19.66	2.69
1, 3	Uniform	−27.06	2.12	−26.96	2.72	−36.19	2.71
	SingleCenter	−27.08	2.17	−27.10	2.54	−34.53	2.52
	DoubleCenter	−28.49	2.05	−28.30	2.81	−32.89	2.81
aver		−21.93	2.11	−21.44	2.62	−25.71	2.65

Table 1. Experiment results with three construction methods

α, β	Instance	diff [%]	Split	DP
			Time	Time
1, 1	Uniform	0.00	2.83	6.83
	SingleCenter	−0.85	2.53	5.53
	DoubleCenter	−0.77	2.63	5.63
1, 2	Uniform	−0.53	2.64	6.64
	SingleCenter	−0.41	2.50	6.50
	DoubleCenter	−0.73	2.69	7.01
1, 3	Uniform	0.00	2.71	6.71
	SingleCenter	−0.17	2.52	5.52
	DoubleCenter	−0.30	2.81	6.81
aver		−0.42	2.65	6.35

Table 2. Experiment results between Exact and GRASP Construction Methods

5.4 The Impact of the Method to Assign Drone Vertices

In this experiment, we evaluate the performance of two methods to assign drone vertices. The experiment results can be found in Table 2.

For using two methods, using the dynamic programming approach helps the proposed algorithm to obtain a slightly better solution than using the greedy approach when the average difference of diff [%] between them is only below 0.42 %

Instances	$\alpha = 1, \beta = 1$			$\alpha = 1, \beta = 2$			$\alpha = 1, \beta = 3$		
	Gap_2	*Improv*	T	Gap_2	*Improv*	T	Gap_2	*Improv*	T
Uniform	−18.58	−4.96	2.09	−23.29	−10.24	2.12	−36.19	−23.91	2.12
SingleCenter	−23.21	−2.58	2.19	−24.68	−4.46	2.17	−34.53	−14.47	2.17
DoubleCenter	−18.37	−4.27	2.07	−19.66	−5.78	2.05	−32.89	−19.55	2.05
aver	−20.05	−3.94	2.12	−22.54	−6.83	2.11	−34.54	−19.31	2.11

Table 3. Experiment results with various the values of α and β

α, β	Instances	Murray et al.			Agatz et al.		
		better	equal	worse	better	equal	worse
1, 1	Uniform	81	10	9	65	12	22
	SingleCenter	83	3	14	67	10	23
	DoubleCenter	85	5	10	60	24	26
1, 2	Uniform	73	0	27	26	42	32
	SingleCenter	79	0	21	51	14	35
	DoubleCenter	66	0	34	45	9	44
1, 3	Uniform	72	0	28	18	64	18
	SingleCenter	79	0	21	21	40	39
	DoubleCenter	77	1	22	16	70	14
Sum		695	19	186	369	286	253

Table 4. Comparisons with two Murray et al.'s and Agatz et al.'s algorithms

(see Table 2). It shows that the greedy approach is also good for assigning drone vertices. More interestingly, while using a dynamic program consumes more time for large instances, the greedy approach spends less time on these cases. Thus, to balance solution quality and running time, using the greedy approach is a suitable choice, especially for large instances.

5.5 The Impact of the Improvement Phase

In this experiment, we evaluate the performance of the improvement phase in developing our solution quality. The experiment results can be found in Table 3.

It is shown that the difference in the average gap between the construction and improvement phases is from −3.94 % to −19.31 %. The average gap is rather large. It indicates the good efficiency of the improvement phase. However, for the larger instances with up to 100 vertices, our search is quite time-consuming. Hence, to reduce the running time, we can only run the construction phase with a loss of −10.03 % solution quality on the overall average.

5.6 Comparison of DMPD Solutions with DMP Solutions

In this experiment, we compare the results of DMP with the drone and DMP without the drone. The experiment results can be seen in Table 3.

Instances	Best.Sol	Aver.Sol	Gap_1	T
uniform-51-maxradius-60	1 322.29	1 335.04	16.74	0.28
uniform51-maxradius100	1 322.29	1 322.29	16.74	0.08
uniform51-maxradius150	1 322.29	1 325.83	16.74	0.07
uniform52-maxradius60	1 135.77	1 186.17	48.17	0.08
uniform52-maxradius100	1 172.43	1 256.52	52.96	0.06
uniform52-maxradius150	1 172.43	1 215.11	52.96	0.05
uniform53-maxradius100	952.87	1 051.09	18.44	0.08
uniform53-maxradius150	973.23	1 046.33	20.97	0.05
uniform54-maxradius60	1 450.59	1 457.04	56.78	0.07
uniform54-maxradius100	1 433.15	1 447.19	54.90	0.16
uniform54-maxradius150	1 433.15	1 449.20	54.90	0.06
uniform55-maxradius60	1 481.55	1 481.55	33.22	0.06
uniform55-maxradius100	1 444.72	1 513.20	29.90	0.06
uniform55-maxradius150	1 444.72	1 499.50	29.90	0.08
uniform56-maxradius60	1 530.57	1 530.57	50.58	0.14
uniform56-maxradius100	1 512.88	1 524.67	48.84	0.09
uniform56-maxradius150	1 530.57	1 530.57	50.58	0.11
uniform57-maxradius100	885.07	885.07	19.23	0.08
uniform58-maxradius1000	1 291.83	1 318.55	27.42	0.08
uniform58-maxradius150	1 291.83	1 291.83	27.42	0.06
uniform60-maxradius40	1 169.87	1 169.87	26.26	0.05
uniform60-maxradius60	1 162.91	1 162.91	25.51	0.07
uniform60-maxradius100	1 162.91	1 175.47	25.51	0.08
uniform60-maxradius150	1 162.91	1 162.91	25.51	0.06
aver			34.59	34.59

Table 5. Experimental results for DMPD with the range constraint

By using the drone, the proposed algorithm reaches better solutions than by using the truck only. Specifically, the average gap between DMPD's and DMP's solution quality is from $-18.58\,\%$ to $-34.54\,\%$. The large difference shows the important role of the drone in improving the solution quality. In all cases, the waiting time of customers is improved though the speed of the drone and truck is equal. Intuitively, when two vehicles deliver simultaneously, the waiting time of clients certainly is improved.

5.7 Comparison of DMPD with Different Speeds of the Drone

In this experiment, we evaluate the impact of different drone speeds on the results of DMPD. The experiment results can be seen in Table 3.

To evaluate the impact of different drone speeds, the ratio $\rho = \frac{\beta}{\alpha}$ is changed from one to three. For example, the ratio 2 means the drone travels at twice the speed of as truck. The results show that all of the proposed algorithms obtain better solutions

Instances	Best.Sol	Aver.Sol	Gap_1	T
uniform_51_novisit_20_rep_1	1 322.29	1 330.55	53.02	0.25
uniform_51_novisit_20_rep_2	1 135.77	1 206.62	31.44	0.18
uniform_51_novisit_20_rep_3	952.87	1 013.97	10.27	0.52
uniform_51_novisit_20_rep_4	1 433.15	1 440.84	65.85	0.10
uniform_51_novisit_20_rep_5	1 444.72	1 490.37	67.19	0.05
uniform_51_novisit_20_rep_6	1 512.88	1 519.95	75.08	0.12
uniform_51_novisit_20_rep_7	885.07	885.07	2.42	0.06
uniform_51_novisit_20_rep_8	1 345.26	1 345.26	55.68	0.05
uniform_51_novisit_20_rep_9	1 162.91	1 186.46	34.58	0.04
uniform_51_novisit_30_rep_10	1 322.29	1 330.55	53.02	0.10
uniform_51_novisit_30_rep_11	1 135.77	1 222.79	31.44	0.10
uniform_51_novisit_30_rep_12	952.87	1 012.38	10.27	0.20
uniform_51_novisit_30_rep_13	1 433.15	1 453.21	65.85	0.12
uniform_51_novisit_30_rep_14	1 444.72	1 478.96	67.19	0.14
uniform_51_novisit_30_rep_15	1 512.88	1 524.67	75.08	0.06
uniform_51_novisit_30_rep_16	885.07	885.07	2.42	0.05
uniform_51_novisit_30_rep_17	1 162.91	1 170.76	34.58	0.06
uniform_51_novisit_40_rep_18	1 322.29	1 334.32	53.02	0.08
uniform_51_novisit_40_rep_19	1 135.77	1 236.68	31.44	0.10
uniform_51_novisit_40_rep_20	1 038.13	1 068.81	20.14	0.12
uniform_51_novisit_40_rep_21	1 359.89	1 411.81	57.37	0.16
uniform_51_novisit_40_rep_22	1 444.72	1 513.20	67.19	0.07
uniform_51_novisit_40_rep_23	1 512.88	1 518.77	75.08	0.08
uniform_51_novisit_40_rep_24	885.07	885.07	2.42	0.05
uniform_51_novisit_40_rep_25	1 291.83	1 305.19	49.50	0.05
uniform_51_novisit_40_rep_26	1 162.91	1 173.38	34.58	0.05
aver			43.31	0.11

Table 6. Experimental results for DMPD with the INCOMP constraint

when the speed of drone increases. Specifically, the drone with double speed further reduces completion times by at least 2 % points, while the drone 3 times as fast as the truck adds at least another 12 % points improvement on average. Note, that in practice, we expect the drone to reach speeds that are more than the speed of the truck[1]. This means that for realistic drone speeds, our heuristics provide very good solutions.

5.8 Comparison with the Other Algorithms for DMPD

In this experiment, we compare the results of the proposed algorithm with the others [36, 15]. The experiment results can be seen in Table 4. In Table 4, each column includes three subcolumns that have “better”, “equal”, and “worse” labels.

[1] `https://www.droneomega.com/how-fast-do-drones-fly/`

Instances	P. Bouman et al.	Murray et al.	Best.Sol
uniform-1-n5	158.65	227.72	158.65
uniform-2-n5	199.07	199.07	199.07
uniform-3-n5	159.65	174.60	159.65
uniform-4-n5	136.10	181.84	136.10
uniform-10-n5	181.50	181.50	181.50
uniform-11-n6	140.55	171.70	140.55
uniform-67-n20	284.46	368.68	340.04
uniform-77-n50	500.55	677.10	564.70
singlecenter-28-n7	133.05	247.25	216.53
singlecenter-33-n8	141.47	166.90	153.21
singlecenter-93-n100	970.58	1 220.91	970.58
singlecenter-94-n100	985.60	1 497.03	985.60
doublecenter-65-n20	504.41	646.16	579.13
doublecenter-86-n75	895.32	1 310.30	895.32
doublecenter-87-n75	982.08	1 342.53	982.08
doublecenter-93-n100	1 057.31	1 479.56	1 057.31
doublecenter-94-n100	1 100.50	1 532.66	1 100.50
doublecenter-95-n100	1 189.19	1 819.39	1 189.19
doublecenter-96-n100	1 163.53	1 705.61	1 163.53

Table 7. Comparisons of the proposed algorithm with the others for TSPD $(\alpha, \beta = 1, 1)$

The "better", "equal", and "worse" labels mean that our solutions are better, equal, and worse in comparison with the other algorithms.

The experimental results for DMPD in Table 4 show that in 900 instances, our algorithm finds 674 better and 207 worse solutions than Murray et al.'s algorithm [15] while it reaches 369 better and 253 worse solutions than Agatz et al.'s algorithm [34]. We also use the Wilcoxon test to evaluate the hypothesis that the proposed method generates statistically better results than the other methods. The results indicate that the proposed algorithm shows a significant improvement over Murray et al.'s algorithm with a level of significance $\alpha = 0.05$. However, since the p-value was larger than the significance level (0.05), it indicates that no statistically significant improvements are detected between our best solutions with Agatz et al.'s ones. Nevertheless, 655 out of 900 better or the same solutions found in comparison with Agatz et al.'s algorithm is still beneficial.

5.9 Results for Variants of DMPD

In this paper, we consider the problem with additional constraints that are:

- Original DMPD: DMPD without any constraints.
- DMPD with INCOMP: A subset of customers must not be visited by the drone. This constraint can be found in Ha et al. [20, 21].

Instances	P. Bouman et al.	Murray et al.	Best.Sol
uniform-1-n5	228.57	227.72	158.65
uniform-14-n6	170.97	162.55	140.02
uniform-44-n9	239.15	222.10	217.63
uniform-52-n10	209.31	229.01	205.55
singlecenter-14-n6	238.41	104.31	99.82
singlecenter-33-n8	249.88	166.90	153.21
singlecenter-92-n100	912.50	1 312.59	1 161.43
singlecenter-99-n100	695.77	1 059.70	979.62
doublecenter-42-n9	483.48	468.60	480.16
doublecenter-68-n20	558.53	698.35	675.46
doublecenter-69-n20	626.51	730.97	712.96
doublecenter-70-n20	531.49	715.15	703.48
doublecenter-71-n50	921.26	1 053.45	1 062.32
doublecenter-72-n50	776.98	1 169.53	1 105.39
doublecenter-73-n50	697.26	990.38	990.96
doublecenter-74-n50	754.47	1 112.06	754.47
doublecenter-75-n50	876.80	1 164.56	1 209.75
doublecenter-76-n50	916.30	1 107.57	916.30
doublecenter-77-n50	754.65	1 065.48	754.65
doublecenter79-n50	794.50	1 068.85	968.70
doublecenter80-n50	879.11	1 245.45	879.11
doublecenter81-n75	969.61	1 265.04	1 305.23

Table 8. Comparisons of the proposed algorithm with the others for TSPD ($\alpha, \beta = 1, 2$)

α, β	Instances	Our algorithm	
		% Gap_1	T
1, 1	poi-10-x	−12.77	2.09
	poi-20-x	−14.56	2.19
	poi-30-x	−15.85	2.07
	poi-40-x	−22.09	2.12
	aver	−24.51	2.17
1, 2	poi-10-x	−12.77	2.09
	poi-20-x	−14.56	2.19
	poi-30-x	−15.85	2.07
	poi-40-x	−22.09	2.12
	aver	−24.51	2.17
1, 3	poi-10-x	−12.77	2.09
	poi-20-x	−14.56	2.19
	poi-30-x	−15.85	2.07
	poi-40-x	−22.09	2.12
	aver	−24.51	2.17

Table 9. Comparisons with Roberti and Ruthmair's algorithm for small instances

- DMPD with Range: The flying range of the drone is limited. This constraint is mentioned in Ha et al. [20, 21].

The constraints are popular when there are many drone-incompatible customers or the limit of drone's range. In this experiment, the common method is to add penalty values to the cost function for each infeasible solution to solve the problems with additional constraints. Therefore, the proposed algorithm is adjusted to adapt to these constraints. In the construction phase, we try to find a feasible solution. If it finds any feasible one, the solution is an input for the improvement phase. Otherwise, the penalty value is added to the original objective function. The advantage of the penalty technique is a simple implementation. With a tour T, let $V(T)$ be the violation. The violation value $V(T)$ is computed as follows:

- For DMPD with INCOMP:

$$V(T) = \max\{d(T), 0\}.$$

- For DMPD with Range:

$$V(T) = \sum^{<i,j,k>\in DR} \max\{c_{ij} + c_{jk} - 2 \times Rd, 0\}.$$

$d(T)$, Rd are the number of vertices that must not be served by the drone but receiving the drone delivery (violation) and the operation range of the drone, respectively. Solutions are then evaluated according to the weighted fitness function $L^{'}(T) = L(T) + \rho * V(T)$, where ρ is the penalty parameter.

Tables 5 and 6 show the results of DMPD with Range, and DMPD with INCOMP. As we know, the constraints make the problem more difficult. In some cases, finding a feasible solution is a challenge. However, the proposed algorithm finds feasible solutions fast for many cases. The differences in the average gap between DMPD's solution and DMPD with Range and INCOMP are 34.59 %, and 43.51 %, respectively. It indicates that the constraints strongly impact the results.

5.10 Results for TSPD

From Tables 7, 8 and 9, we tested the proposed algorithm for TSPD. Our algorithm still runs well for TSPD, although it was not designed to solve it. In comparison with the metaheuristic algorithms of Murray et al. [15] and Agatz et al. [16], our results are better than Murray et al.'s ones and are comparable with Agatz et al.'s ones in many cases. In addition, we run the proposed algorithm with small instances in [37]. Specifically, the proposed algorithm can find near-optimal solutions for the instances with 40 vertices. In many cases, the exact algorithm in [37] fails to find the optimal solution because of memory and time limits. On the other hand, the proposed algorithm reaches better solutions in these cases.

The algorithms in[16, 15] are executed on the same configuration. Therefore, it is convenient for us to compare the running time exactly. From the experimental results, the running time of the proposed algorithm is comparable with the other algorithms.

6 DISCUSSIONS

In this paper, we propose the hybrid approach between VNS with Tabu and Shaking, as follows. Firstly, the VNS maintains intensification by generating many good locally optimal solutions around the optimal global solution while the Shaking tries to maintain diversification. The combination balances diversification and intensification. Secondly, TS prevents the search from getting trapped into cycles. It helps the search to drive to the optimal global solution.

Considering the experiments, the proposed algorithm is tested with various scenarios: investigate the performance of different construction heuristics, explore the performance of different split heuristics, analyze the impact of drone on solution quality, and compare our solution quality with the previous algorithms. From the experimental results, we conclude:

- DMP's solution quality is relatively important to generate an initial solution for DMPD. In some methods, the metaheuristic (GRASP with VNS) obtains better solutions than the others. In small instances, the exact algorithm can be used to reach the optimal solutions for DMP. However, in most cases, the exact solutions for DMP cannot help the proposed algorithm find better solutions for DMPD while it consumes more time. The GRASP with VNS is the best choice in terms of solution quality and running time.
- The split step takes an important role in the final result. Using dynamic programming can receive a very slightly better than the greedy approach but it consumes much time for large instances. On the other hand, the greedy approach balances between solution quality and running time well. Therefore, the first strategy for decreasing the running time is using the greedy method with a loss of $-10.03\,\%$ solution quality on average.
- The results demonstrate that drone supports us in improving solution quality in comparison with truck only. The higher the speed of drone is, the more the total waiting time of clients decreases. In addition, the proposed algorithm finds feasible solutions for DMPD with constraints. The experimental results show that the constraints strongly affect the solution quality. It implies that the problem becomes much more complex when the constraints are involved.
- The algorithm finds better solutions than Murray et al. [15], and Agatz et al. [16] in the case of DMPD. The proposed algorithm is also tested with TSPD's instances. The results show that the proposed algorithm is comparable with the other algorithms [36, 15]. In addition, the proposed algorithm can find nearly

optimal solutions for the instances with 40 vertices. In many cases, the proposed algorithm finds better solutions in these cases, while Roberti and Ruthmair's algorithm [37] fails because of memory or time limits. Therefore, the proposed algorithm can be applied to TSPD well, though, it is not designed to solve it.

7 CONCLUSIONS

In this paper, we introduce DMPD. The main contribution is to propose a metaheuristic algorithm that balances intensification and diversification for solving DMPD. In the first step, a solution without the drone is generated by some heuristics. Then, it is an input of split procedure to convert it into DMPD-solution. After that, it is improved by the combination of Variable Neighborhood Search (VNS) and Tabu Search (TS). We implemented the algorithm on the benchmark dataset to compare it to several state-of-the-art metaheuristics. The proposed algorithm obtains better solutions in comparison with the other algorithms in many cases. Our algorithm is also tested with TSPD. The results show that our solutions can be compared with the previous algorithms, and the proposed algorithm can find near-optimal solutions for instances with up to 40 vertices in a short time. However, the running time of the algorithm can be improved to meet practical situations. In addition, applying to some different neighborhoods will be investigated in future research. Moreover, some other aspects will be considered to make the problem a more practical situation. In the first aspect, truck uses gasoline to move. It can run a long distance. Therefore, we do not consider the energy consumption of truck in this paper. Otherwise, the drone needs to return to the truck after each delivery to recharge energy and its range is limited because of battery-powered engines. Nevertheless, we assume that the drone always recharges energy when needed. In the second aspect, multiple drones and trucks deliver packets at the same time. Considering these aspects will be our aim in future work.

Acknowledgements

This research was supported by the Hanoi University of Science and Technology under grant number T2021-PC-021.

REFERENCES

[1] Poikonen, S.—Golden, B.—Wasil, E. A.: A Branch-and-Bound Approach to the Traveling Salesman Problem with a Drone. INFORMS Journal on Computing, Vol. 31, 2019, No. 2, pp. 335–346, doi: 10.1287/ijoc.2018.0826.

[2] Burgess, M.: DHL's Parcelcopter Drone Can Make Drops Quicker Than a Car. WIRED UK, 2016, https://www.wired.co.uk/article/dhl-drone-delivery-germany.

[3] Popper, B.: Hydrogen Fuel Cells Promise to Keep Drones Flying for Hours. The Verge, 2015, https://www.theverge.com/2015/12/15/10220456/intelligent-energy-hydrogen-fuel-cell-drone.

[4] Ajam, M.—Akbari, V.—Salman, F. S.: Minimizing Latency in Post-Disaster Road Clearance Operations. European Journal of Operational Research, Vol. 277, 2019, No. 3, pp. 1098–1112, doi: 10.1016/j.ejor.2019.03.024.

[5] Avci, M.—Avci, M. G.: A GRASP with Iterated Local Search for the Traveling Repairman Problem with Profits. Computers and Industrial Engineering, Vol. 113, 2017, pp. 323–332, doi: 10.1016/j.cie.2017.09.032.

[6] Ban, H. B.—Nguyen, D. N.: Improved Genetic Algorithm for Minimum Latency Problem. Proceedings of the 1st Symposium on Information and Communication Technology (SoICT '10), ACM, 2010, pp. 9–15, doi: 10.1145/1852611.1852614.

[7] Ban, H. B.—Nguyen, K.—Ngo, M. C.—Nguyen, D. N.: An Efficient Exact Algorithm for the Minimum Latency Problem. Progress in Informatics, Vol. 10, 2013, pp. 167–174, doi: 10.2201/NiiPi.2013.10.10.

[8] Ban, H. B.—Nguyen, D. N.: A Meta-Heuristic Algorithm Combining Between Tabu and Variable Neighborhood Search for the Minimum Latency Problem. Fundamenta Informaticae, Vol. 156, 2017, No. 1, pp. 21–41, doi: 10.3233/FI-2017-1596.

[9] Ban, H. B.: A Metaheuristic for the Delivery Man Problem with Time Windows. Journal of Combinatorial Optimization, Vol. 41, 2021, No. 4, pp. 794–816, doi: 10.1007/s10878-021-00716-2.

[10] Blum, A.—Chalasani, P.—Coppersmith, D.—Pulleyblank, B.—Raghavan, P.—Sudan, M.: The Minimum Latency Problem. Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing (STOC '94), 1994, pp. 163–171, doi: 10.1145/195058.195125.

[11] Chaudhuri, K.—Godfrey, B.—Rao, S.—Talwar, K.: Paths, Trees, and Minimum Latency Tours. 44th Annual IEEE Symposium on Foundations of Computer Science, 2003, pp. 36–45, doi: 10.1109/SFCS.2003.1238179.

[12] Dewilde, T.—Cattrysse, D.—Coene, S.—Spieksma, F. C. R.—Vansteenwegen, P.: Heuristics for the Traveling Repairman Problem with Profits. Computers and Operations Research, Vol. 40, 2013, No. 7, pp. 1700–1707, doi: 10.1016/j.cor.2013.01.003.

[13] Goemans, M.—Kleinberg, J.: An Improved Approximation Ratio for the Minimum Latency Problem. Mathematical Programming, Vol. 82, 1998, No. 1-2, pp. 111–124, doi: 10.1007/BF01585867.

[14] Beraldi, P.—Bruni, M. E.—Laganà, D.—Musmanno, R.: The Risk-Averse Traveling Repairman Problem with Profits. Soft Computing, Vol. 23, 2019, No. 9, pp. 2979–2993, doi: 10.1007/s00500-018-3660-5.

[15] Murray, C. C.—Chu, A. G.: The Flying Sidekick Traveling Salesman Problem: Optimization of Drone-Assisted Parcel Delivery. Transportation Research Part C: Emerging Technologies, Vol. 54, 2015, pp. 86–109, doi: 10.1016/j.trc.2015.03.005.

[16] Agatz, N.—Bouman, P.—Schmidt, M.: Optimization Approaches for the Traveling Salesman Problem with Drone. Transportation Science, Vol. 52, 2018, No. 4, pp. 965–981, doi: 10.1287/trsc.2017.0791.

[17] Bouman, P.: Code for the Traveling Salesman Problem with Drone. 2018, `https://github.com/pcbouman-eur/Drones-TSP`.

[18] de Freitas, J. C.—Penna, P. H. V.: A Variable Neighborhood Search for Flying Sidekick Traveling Salesman Problem. International Transactions in Operational Research, Vol. 27, 2019, No. 1, pp. 267–290, doi: 10.1111/itor.12671.

[19] Poikonen, S.—Wang, X.—Golden, B.: The Vehicle Routing Problem with Drones: Extended Models and Connections. Networks, Vol. 70, 2017, No. 1, pp. 34–43, doi: 10.1002/net.21746.

[20] Ha, Q. M.—Deville, Y.—Pham, Q. D.—Hà, M. H.: On the Min-Cost Traveling Salesman Problem with Drone. Transportation Research Part C: Emerging Technologies, Vol. 86, 2018, pp. 597–621, doi: 10.1016/j.trc.2017.11.015.

[21] Ha, Q. M.—Deville, Y.—Pham, Q. D.—Hà, M. H.: A Hybrid Genetic Algorithm for the Traveling Salesman Problem with Drone. Journal of Heuristics, Vol. 26, 2020, No. 2, pp. 219–247, doi: 10.1007/s10732-019-09431-y.

[22] Daknama, R.—Kraus, E.: Vehicle Routing with Drones. CoRR, 2017, doi: 10.48550/arXiv.1705.06431.

[23] Dorling, K.—Heinrichs, J.—Messier, G. G.—Magierowski, S.: Vehicle Routing Problems for Drone Delivery. IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol. 47, 2017, No. 1, pp. 70–85, doi: 10.1109/TSMC.2016.2582745.

[24] Kim, S.—Moon, I.: Traveling Salesman Problem with a Drone Station. IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol. 49, 2019, No. 1, pp. 42–52, doi: 10.1109/TSMC.2018.2867496.

[25] Schermer, D.—Moeini, M.—Wendt, O.: The Traveling Salesman Drone Station Location Problem. In: Le Thi, H. A., Le, H. M., Pham Dinh, T. (Eds.): Optimization of Complex Systems: Theory, Models, Algorithms and Applications (WCGO 2019). Springer, Cham, Advances in Intelligent Systems and Computing, Vol. 991, 2020, pp. 1129–1138, doi: 10.1007/978-3-030-21803-4_111.

[26] Mbiadou Saleu, R. G.—Deroussi, L.—Feillet, D.—Grangeon, N.—Quilliot, A.: An Iterative Two-Step Heuristic for the Parallel Drone Scheduling Traveling Salesman Problem. Networks, Vol. 72, 2018, No. 4, pp. 459–474, doi: 10.1002/net.21846.

[27] Wang, X.—Poikonen, S.—Golden, B.: The Vehicle Routing Problem with Drones: Several Worst-Case Results. Optimization Letters, Vol. 11, 2017, No. 4, pp. 679–697, doi: 10.1007/s11590-016-1035-3.

[28] Salehipour, A.—Sörensen, K.—Goos, P.—Bräysy, O.: Efficient GRASP + VND and GRASP + VNS Metaheuristics for the Traveling Repairman Problem. 4OR, Vol. 9, 2011, No. 2, pp. 189–209, doi: 10.1007/s10288-011-0153-0.

[29] Johnson, D. S.—McGeoch, L. A.: The Traveling Salesman Problem: A Case Study in Local Optimization. Chapter 8. In: Aarts, E. H. L., Lenstra, J. K. (Eds.): Local Search in Combinatorial Optimization. John Wiley & Sons Ltd., 1997, pp. 215–310.

[30] Feo, T. A.—Resende, M. G. C.: Greedy Randomized Adaptive Search Procedures. Journal of Global Optimization, Vol. 6, 1995, No. 2, pp. 109–133, doi:

10.1007/BF01096763.

[31] Belošević, I.—Ivić, M.: Variable Neighborhood Search for Multistage Train Classification at Strategic Planning Level. Computer-Aided Civil and Infrastructure Engineering, Vol. 33, 2018, No. 3, pp. 220–242, doi: 10.1111/mice.12304.

[32] Hansen, P.—Mladenović, N.—Todosijević, R.—Hanafi, S.: Variable Neighborhood Search: Basics and Variants. EURO Journal on Computational Optimization, Vol. 5, 2017, No. 3, pp. 423–454, doi: 10.1007/s13675-016-0075-x.

[33] Mladenović, N.—Hansen, P.: Variable Neighborhood Search. Computers and Operations Research, Vol. 24, 1997, No. 11, pp. 1097–1100, doi: 10.1016/S0305-0548(97)00031-2.

[34] Bouman, P.: TSP-D-Instances. 2015, https://github.com/pcbouman-eur/TSP-D-Instances.

[35] Appendix-DMPD.pdf. https://drive.google.com/file/d/16SmLpCUs_nekhO5_yaCkWoiPWgpBaZB5/view?usp=sharing.

[36] Bouman, P.—Agatz, N.—Schmidt, M.: Dynamic Programming Approaches for the Traveling Salesman Problem with Drone. Networks, Vol. 72, 2018, pp. 528–542, doi: 10.1002/net.21864.

[37] Roberti, R.—Ruthmair, M.: Exact Methods for the Traveling Salesman Problem with Drone. Transportation Science, Vol. 55, 2021, No. 2, pp. 315–335, doi: 10.1287/trsc.2020.1017.

Ha-Bang Ban received his Ph.D. in computer science at the Hanoi University of Science and Technology (HUST), Vietnam in 2015. He is currently the Lecturer at the School of Information and Communication Technology (SoICT), HUST, Vietnam. His research interests include algorithms, graphs, optimization, logistics, etc. He has published many publications in peer-reviewed international journals and conferences.

Hai-Dang Pham received his engineering diploma in information technology from the Hanoi University of Science and Technology (HUST), Vietnam, in 1995 and his Ph.D. in computer science from École Pratique des Hautes Études (EPHE), France, in 2011. He is currently the Senior Lecturer at the School of Information and Communication Technology (SoICT), HUST, Vietnam. His current research interests include algorithms, parallel and distributed simulation, multi-agent based simulation and high performance computing.

Computing and Informatics, Vol. 42, 2023, 1213–1232, doi: 10.31577/cai_2023_5_1213

LOCATION ESTIMATION FROM AN INDOOR SELFIE

Mengqi Du, Yue Zhang

College of Computer Science and Technology
Zhejiang University of Technology
Hangzhou, China
e-mail: mengqidu@foxmail.com, 1111712013@zjut.edu.cn

Jianhua Zhang

School of Computer Science and Engineering
Tianjin University of Technology
Tianjin, China
e-mail: zjh@ieee.org

Honghai Liu

School of Mechanical Engineering and Automation
Harbin Institute of Technology
Shenzhen, China
e-mail: honghai.liu@icloud.com

Abstract. With the development of social networks and hardware devices, many young people have post a lot of high definition v-logs containing selfie images and videos to commemorate and share their daily lives. We found that the reflected image of corneal position in the high definition selfie image has been able to reflect the position and posture of the selfie taker. The classic localization works estimating the position and posture from a selfie are difficult because they lack the knowledge of the environment. The corneal reflection images inherently carry information about the surrounding environment, which can reveal the location, posture and even height of the selfie taker. We analyze the corneal reflection imaging process in the selfie scenario and design a validation experiment based on this process to

estimate the pose of the selfie in several scenarios to further evaluate the leakage of the pose information of the selfie taker.

Keywords: Corneal imaging system, location estimation, privacy disclosure, selfie, social network

Mathematics Subject Classification 2010: 65-D19

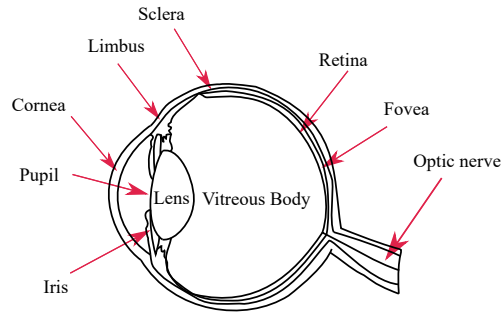
1 INTRODUCTION

Posting v-logs documenting and sharing life on social network sites is very popular among young people. However, selfie photos and videos often reveal the shooting time, environment, location, and even the habits of the shooter. In addition, as the quality of camera imaging has improved, the corneal reflection of the photographer's surroundings is much clearer than before. The existing works have demonstrated that corneal reflecting image can show the gaze and intention of the subject, and even analyze the environment in which the subject is located.

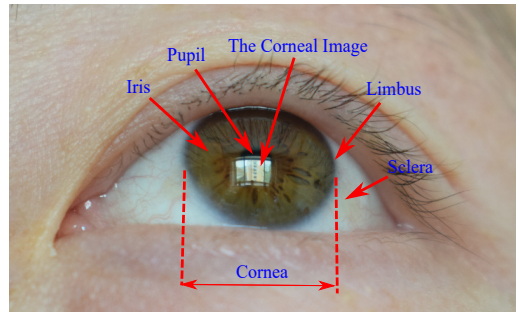
After a lot of observations, we found that the corneal reflecting images can even reveal the position and posture of the selfie taker. These information may reveal the selfie shooting habit even the height of the body. In this paper, we analyze the corneal imaging process during selfie, and experimentally verify and analyze that there is a certain correspondence between the corneal reflecting images and the position, posture of the selfie taker.

The cornea is a transparent semi-ellipsoidal structure located at the very front of the eye and plays a protective role for the eye, as shown in Figure 1 a). Incident rays shoot through the cornea, pass through the pupil, lens, and vitreous body to reach the fovea on the retina, where the light signal is converted into an electrical signal then transmitted by the optic nerve to the visual center of the brain, so that the body can perceive its surroundings visual information. The corneal surface is covered with a thin, reflexible tear film, when the incident rays pass through the cornea, a small amount of light will be reflected by the tear film. These reflected light can be captured by the human eye or optical device such as a camera, then form an image, this image is called corneal image (CI). The incident light – the corneal reflective surface – the reflected light capture device together form the corneal imaging system (CIS).

Since a CI can reflect the human surroundings, the CIS has extensive research and application prospects in the fields of human-machine interaction, computer graphics, disaster rescue, indoor and outdoor localization, and so on. However, there are still difficulties in the research of CIS. Because the cornea is a transparent structure, only a limited amount of incident light can be reflected back into the camera, creating a CI with little details and texture. In addition, the CI is always polluted by the color and texture of the iris in the eye image.



a) The anatomical view of eye



b) The frontview of eye

Figure 1. The structure of eye

Despite the poor imaging quality, however, it has been found from daily observation that the CIS is sensitive to high-light scenes: the high-light incident rays will be projected as the brightness pattern on the corneal surface, as shown in Figure 1 b). Therefore, many works based on the CIS require additional pieces of equipment, such as an infrared camera to determine the location of the high-light pattern in the CI, or a reliable light source, such as a bright, high-contrast light source as input to improve the quality of CI.

When taking indoor selfies, in order to obtain better imaging quality, the selfie taker often chooses a scene with better light distribution, such as facing a window or a screen being projected, so that an indoor high-light element can be projected as a bright pattern on the cornea. The projected pattern shape will change along the corneal pose transformation, we then can estimate the pose variation of the photographer. As an innovative and challenging work, in this paper, we will test this innovative hypothesis from both theoretical and experimental aspects.

2 RELATED WORK

2.1 Monocular Camera Based Localization

The monocular RGB camera positioning works enlighten our work. By recording the center displacement of the camera and simultaneously extracting and matching the affine invariant features between frames (e.g., SIFT [1], corner point [2]), or with the help of the scene geometry knowledge [3] (e.g. parallel lines, surfaces), to achieve the estimation of the change of the target object's pose. Monocular RGB cameras are widely used in AR [4, 5], visual SLAM [6, 7, 8], 3D reconstruction [9, 10], and other fields. The way the camera estimates the scene depth relies heavily on the feature matching between frames. The localization accuracy will be poor when the input light is weak. To improve the localization accuracy of monocular cameras, existing works try to find effective matching features among the matchable feature clusters with algorithms such as RANSAC [11], or artificially introduce distinctive markers in the scene, such as AprilTag [12], ARToolkit [13] to improve the feature extraction and matching accuracy. However, since the CI contain little texture, and details and lacks a sufficient number of effective feature points, the above methods cannot be applied in the localization method of this paper.

2.2 The CIS Researches and Applications

The subject's behavior and awareness can be inferred from the environmental images reflected from the cornea. This is the force that drives the research of the CIS. In order to analyze the light distribution around the subject, Tsumura et al. [14] calculate the source of light by analyzing the light spot reflected from the cornea to reconstruct the face model. Nishino and Nayar [15] view the cornea as a light probe to percept the light distribution of the scene then relighting the given 3D face.

To improve the imaging quality of CIS, Wang et al. [16] propose a CIS separation algorithm with two eye images as input. Nitschke and Nakazawa [17], based on the corneal and eye models through the super-resolution has proposed a method for CI enhancement.

Nishino and Nayar [18] reference and extend the work of Swaminathan et al. [19] to the field of CIS, and fully explain the relationship between scenes, corneas and cameras involved in CIS. As a complementary work to Nishino and Nayar [18], Nitschke et al. [20] propose a calibration method for CIS based on an infrared camera and an LED light array. Based on the above theory of CIS, Suda et al. [21] propose an algorithm for matching the CI to the scene in order to avoid the calibration process. Nakazawa et al. [22] propose a gaze-tracking algorithm based on the CIS using a bendable LED dot matrix system assisted by an infrared camera. Lander et al. [23] propose a work for computing 3D gaze from the 2D gaze with the help of infrared cameras and scene cameras. Ohshima et al. [24] try to match the CI with the scene pictures in the database by neural network, showing the possible privacy security risk and the prospect of human-machine interaction for CIS. Du et al. [25]

propose a gaze tracking method based on CIS with the help of an AprilTag marker, which makes it possible to use gaze as an AR interaction method.

As a catadioptric imaging system, the CIS has been developed in the past two decades, and its main objectives are focused on assisting gaze-tracking and the analysis and recognition of reflection scenes. As an important component of CIS, the posture of the cornea is critical to the imaging process, and in addition, the posture of the cornea is often closely related to the posture of the person. However, there is less existing related work. By analyzing the relationship between scene, corneal pose and head pose, we try to verify the feasibility of CIS-based indoor localization and show the possibility of privacy leakage risk of selfies in social network in this paper.

3 THEORETICAL ANALYSIS OF CORNEAL IMAGING PROCESS DURING SELF-TIMER

The imaging process of the CIS is shown in Figure 2. The incident ray $\boldsymbol{v}_i \in V_I$ is reflected by the cornea as reflected light $\boldsymbol{v}_r \in V_R$, and part of the reflected ray set $V_C \subset V_R$ can be captured by the camera to participate in the corneal imaging process. When the subject is indoors and facing the high-light L, the set of incident corneal rays $V_H \subset V_I$ from the L, of which the partially reflected rays $V_{HR} \subset V_C$, are captured by the camera and become the bright pattern of the CI. The rest of the CI comes from the partially reflected rays $V_{SR} \subset V_C$ which are the reflected rays of incident rays V_S in the scene.

There are two projection processes in CIS. Firstly, the incident rays V_I are projected to the corneal surface. Secondly, the reflected rays V_C are projected to the camera imaging plane. The corneal imaging process can be expressed as Equation (1).

$$I = P_c V_I, \tag{1}$$

where P_c is the projection matrix, which can be expressed as Equation (2):

$$P_c = K_c \left[R_c | \boldsymbol{t}_c \right], \tag{2}$$

where

$$\boldsymbol{t}_c = -R_c \tilde{C}_c,$$

where K_c, R_c and $\tilde{C}_c$ are the intrinsic parameter, rotation matrix and the inhomogeneous coordinate of the camera center position, respectively.

As the extrinsic camera parameters, R_c and $\tilde{C}_c$ described the posture and position of the CIS in the world coordinate system O_w. Without loss of generality, the head pose can be considered to represent R_c. In this paper, the subject posture and position we try to estimate can be represented by R_c as well as $\tilde{C}_c$.

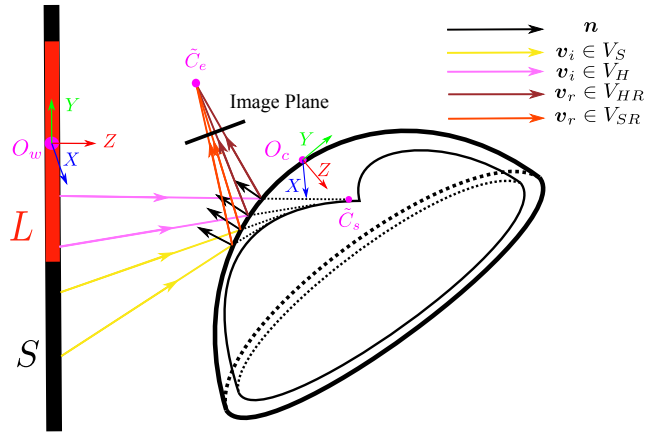


Figure 2. Imaging process of CIS. The incident ray set V_I is reflected by the cornea, and a portion of the reflected ray set V_C is captured by the camera to form a CI.

3.1 Intrinsic Parameter of CIS

The intrinsic parameter K_c of the CIS can be expressed as

$$K_c = P_e P_s R_g,$$

where P_e is the projection matrix of the camera, which takes the form $P_e = K_e\,[R_e|\boldsymbol{t}_e]$, where K_e is the intrinsic parameter of the camera, $\boldsymbol{t}_e = -R_e\tilde{C}_e$, R_e and $\tilde{C}_e$ are the extrinsic parameters of the camera.

The form of K_e is as in Equation (3), without loss of generality, assuming that the camera pixels are square $f_x = f_y = f$, skew parameter $s = 0$, and the camera center is at the center of the imaging plane $c_x = c_y = 0$.

Then K_e in the projective transformation mainly scales the incident light and reduces the dimension.

$$K_e = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3}$$

R_e is the rotation matrix of the camera, and $\tilde{C}_e$ is the position of the CIS represented in inhomogeneous coordinate form in the world coordinate O_w.

The Z component of $\tilde{C}_e$ affects the size of the face in the image, which is mainly affected by the subject's shooting habits and arm length, while the X, Y components mainly affect the position of the bright pattern in the CI. The $\tilde{C}_e$ has less influence on the position of the bright pattern in the CI when the face area is often centered in the selfie.

In order to estimate the effect of R_e on the CIS when taking a selfie, we invite 9 subjects sit in front of the screen in turn, complete the selfie with the front lens of the phone and keep the phone position still, and then take the AprilTag marker photo which displayed on the screen with the rear lens, calculate the orientation of the phone at this moment, and repeat five times for each person, record and count the data, as shown in Figure 3. It can be seen that the camera imaging plane is basically parallel to the screen plane, and R_e has little effect on the CIS.

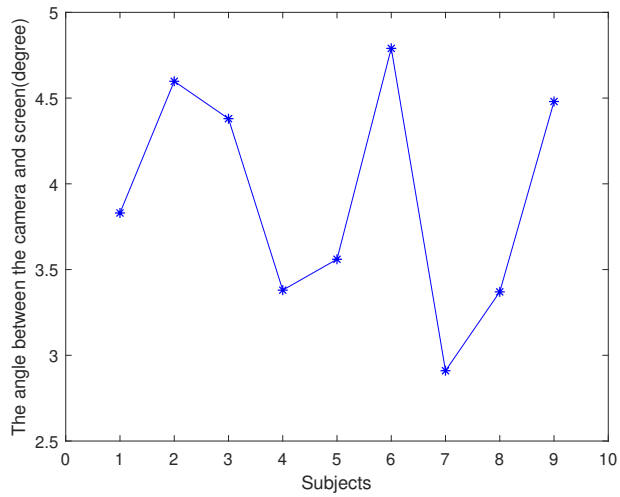


Figure 3. Camera pose distribution when taking selfies of different subjects

$R_g = \{yaw, pitch, roll\}$ is the rotation of the eyeball during corneal reflection imaging, *yaw*, *pitch*, and *roll* are the rotation angles of the eyeball around the Y, X, and Z axes under the coordinate system O_c, respectively. The eyeball rotation affects the pose of the corneal reflective surface P_s.

We refer to the work of Nishino and Nayar [18] to model the corneal geometry structure and analyze the influence of the corneal reflective surface posture P_s during the corneal imaging process, as shown in Figure 4. The corneal geometry can be described as

$$S(t, \theta) = (S_x, S_y, S_z) = (\lambda \cos\theta, \lambda \sin\theta, t)\,,$$

where $t \in [0, 2.18]$, $\theta \in [0, 2\pi]$ and

$$\lambda = \sqrt{-pt^2 + 2Rt}.$$

Based on the anatomical work of Kaufman and Alm [26], the shape of the cornea is found to be essentially the same in different adults. The radius of curvature at the vertex $R = 7.8\,\text{mm}$, the mean eccentricity $e = 0.5$, and $p = 1 - e^2$. The tangent

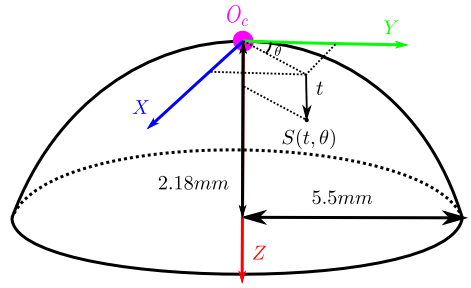


Figure 4. Modeling the geometric structure of the cornea. Corneal height was 2.18 mm, limbus as the outermost edge, which can be considered as a circle, has a radius of 5.5 mm.

$\boldsymbol{\tau}(t,\theta) = \left[\nabla_{S_x}, \nabla_{S_y}, \nabla_{S_z}\right]^T$ to any point $S(t,\theta)$ on X, Y, Z

$$\nabla_{S_x} = \frac{\cos\theta}{2\sqrt{-1.5t+15.6}} - \sin\theta\sqrt{-0.75t^2+15.6t},$$

$$\nabla_{S_y} = \frac{\sin\theta}{2\sqrt{-1.5t+15.6}} + \cos\theta\sqrt{-0.75t^2+15.6t},$$

$$\nabla_{S_z} = 1.$$

For any point $S(t,\theta)$ on the corneal surface, its normal vector $\boldsymbol{n}(t,\theta) = [F_x, F_y, F_z]^T$. The $\boldsymbol{n}(t,\theta)$ and $\boldsymbol{\tau}(t,\theta)$ satisfy

$$\boldsymbol{n}^T(t,\theta)\cdot\boldsymbol{\tau}(t,\theta) = F_x\nabla S_x + F_y\nabla S_y + F_z\nabla S_z = 0.$$

The projection of an incident ray on the corneal surface can be expressed by Equation (4):

$$P_s = K_s\left[R_s|\boldsymbol{t}_s\right], \tag{4}$$

where $\boldsymbol{t}_s = -R_s\tilde{C}_s$, K_s as the intrinsic parameter has similar values for different $\tilde{C}_s$. R_s and $\tilde{C}_s$ are the extrinsic parameters of P_s. The rotation matrix R_s can be obtained from $\boldsymbol{n}(t,\theta)$ by the Rodriguez rotation formula.

$\tilde{C}_s$ is the position of the optical center in O_w for a single corneal reflecting surface expressed in inhomogeneous form. $\tilde{C}_s$ is determined jointly by the incident light $\boldsymbol{v}_i$ and the corneal surface $S(t,\theta)$ involved in the reflection. Different surface involved in the corneal imaging process will have its viewpoint. All viewpoints will form the viewpoint trajectory envelope:

$$V(t,\theta,r) = S(t,\theta) + r\boldsymbol{v}_i(t,\theta),$$

where r is the distance between $S(t,\theta)$ and $\tilde{C}_s$, which can be obtained by

$$detJ(V(t,\theta,r)) = 0,$$

where J is the Jacobi matrix.

The factors act on P_s include three points, namely R_g, R_c and $\tilde{C}_c$. Eye rotation R_g has the same effect on R_s as the subject's head posture change R_c, but when R_g changes, it is accompanied by the corresponding change in the position of the cornea, iris, etc. in the eye. Therefore, we can analyze R_g by the position of cornea in the eye image, and then detach the effect of R_g from P_s to avoid its influence on the estimation of R_c. So far, the CI is unique for a given position R_c, $\tilde{C}_c$, and a given R_g, in the same scene. In fact, shooting CI is often accompanied by the freedom of eyeball rotation, so that for any R_c, $\tilde{C}_c$ corresponds to a set of CIs with different projection bright patterns and iris positions.

3.2 Pose and Position Estimation of CIS

The image quality of CI acquired during self-photography is poor. In addition, the high-light L may be projected incompletely on the corneal reflective surface due to the effect of eyeball rotation, even though the subject is facing L. These two aspects lead to difficulties in solving the exact R_c, $\tilde{C}_c$ by stripping the projection matrix K_c in Equation (1).

We have known that there exists a set of CIs corresponding to the subject in any one of the poses R_c and position $\tilde{C}_c$. These CIs are not the same due to the influence of P_e, R_g. As described in Section 3.1, P_e has less effect on the selfie imaging process, and in our experiments, we have focused on the role of R_g on indoor localization.

We try to test our work in different high-light scenes and take a set of CIs with different R_g for the given subject's pose and position. We will use these CIs with various positions and poses to train the neural network and ensure that the neural network can predict the positions and poses of the subjects in the CIs.

In addition, the size of the high-light L is another factor that affects the positioning performance. Different sizes of L will have different positioning accuracy and effective positioning range. Based on the above hypothesis, we try to design an experiment to verify our idea and also estimate the accuracy performance under different high-light source sizes.

To observe the performance of localization, we find the effective indoor ground range S, and divide S into 64 equal regions, each region is centered at $C_i \in C$. In the data acquisition process, we let the subject take a selfie in R_{ci} pose at three different locations in region i and record the location data $C_{ci} = \{C_{ix}, C_{iy}, H_s\}$ containing the subject's height H_s, the pose data R_{ci} and the eye part of the selfie image. Next, we record the test set data at the center C_i in the same data acquisition manner.

We use eye photos and high-light size of the D_{Train} as input and pose, position information as output to complete the training of VGG16 network. Using the eye photos and high-light size from the D_{Test} as the input of the trained VGG16, we then compare the output of VGG16 network with the ground truth data to

verify the feasibility of localization by CIs and analyze the effective localization range.

The indoor space S includes some indistinguishable regions, and we exclude the above locations from the effective space S in order to speed up the convergence of the training network and reduce the possibility of over-fitting. For locations very close to the high-light L, the bright pattern of the high-light projection completely occupies the CI, and in this case, the shape of the bright pattern does not reflect the position of the subject. When the subject is far away from L, the shape of the bright pattern does not reflect the change of pose and position, both of which are indistinguishable regions.

4 POSITION AND POSE ESTIMATION PERFORMANCE

We looked for 60 volunteers with different heights, ages and genders as subjects. We measured the height of the subject as shown in Figure 5. We conducted experiments in three experimental fields $S = \{S_1, S_2, S_3\}$ with three different high-light areas $L = \{L_1, L_2, L_3\}$, respectively, to observe the effect of different high-light areas on the localization. The sizes of L_1, L_2, L_3 are $280 \times 160\,\mathrm{cm}^2$, $130 \times 120\,\mathrm{cm}^2$ and $55 \times 33\,\mathrm{cm}^2$. Each experimental field was divided into 64 regions with 90 cm, 50 cm and 25 cm sides, as described in Section 3.2.

For subject s, the head pose is $R_{cs} = \{yaw, pitch, 0°\}$, where yaw and $pitch$ are the rotation angles of s around the Y and X axes in world coordinates, respectively. The yaw angle range is

$$yaw \in \{-60°, -45°, -30°, -15°, 0°, 15°, 30°, 45°, 60°\}$$

and pitch angle range is

$$pitch \in \{-30°, 0°, 30°\}.$$

At different R_c, $\tilde{C}_c$, the subject s holds a camera and takes a selfie video with different eyeball rotations for at least 15 seconds at 60 fps, 4 k resolution to collect data from the training and test dataset. To ensure that the eye image was less affected by the subject's motion and pose, we asked the subjects to maintain as stable a pose as possible, and also asked them to rotate their eyes at will when they could see the high-light plane to ensure the presence of a more complete light spot in the CI. We then segmented the video into images and split the eye images by frame.

Nevertheless, there were some unusable images, especially when subjects stand in the edge regions of S and were asked to collect experimental data in extreme poses (e.g., $yaw = 60°$, $pitch = 30°$). As in Figure 6, when the eye image has severe motion blur, or when the reflected spot is heavily obscured, such an eye image is invalid and needs to be removed from the dataset. After the experimental data collection and collation process, we get 34 209 294 valid images from 41 472 000 eye images in three experimental spaces, accounting for 82.48 %. The efficient distribution of data

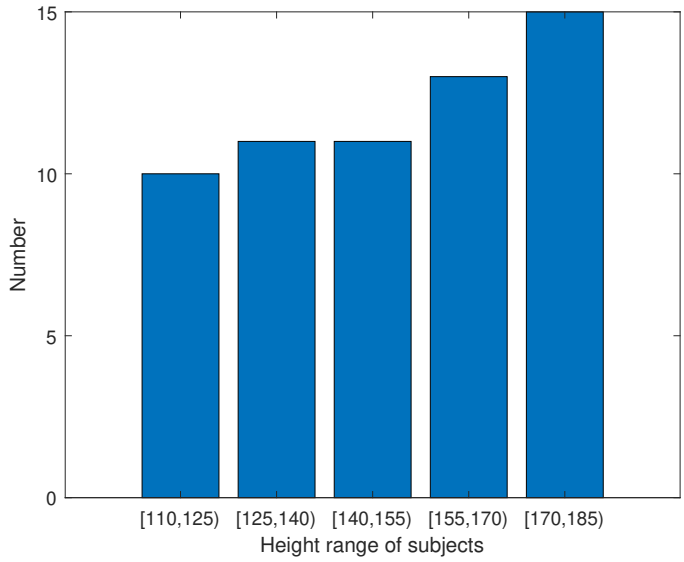
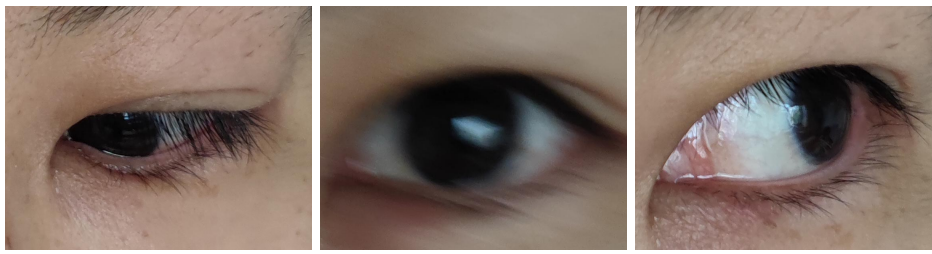


Figure 5. Height distribution of the subjects

in each region is shown in Figure 7. We can see that the closer to the center of S, the more efficient the data are, and the least efficient the data are located at the four corners.



a) Light spot is heavily covered b) Severe motion blur c) Limited spot reflected

Figure 6. Types of video frames that need to be removed from the dataset

After collecting the dataset, we trained and tested the VGG16 network according to Section 3.2.

During the experiment, we find that the key factor affecting the localization accuracy is the size of the spot. Small spots require a wide range of subject's movement and position changes to produce significant deformation. The vertical distance from the measured object to the high-light plane and the orientation angle of the

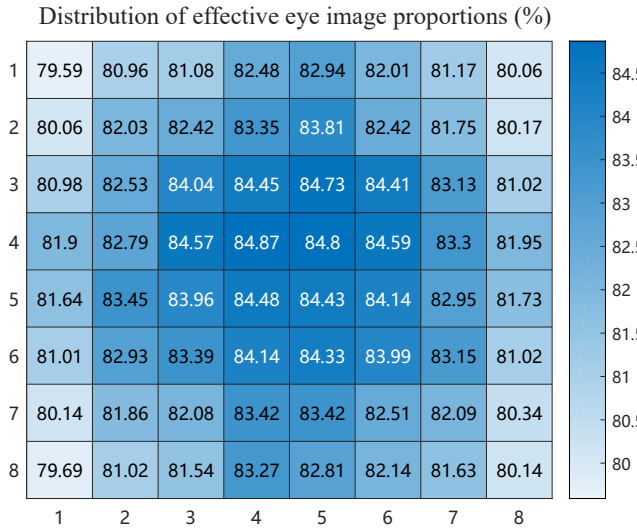


Figure 7. Distribution of available data in each region of the experimental space S as a percentage of the collected data

measured object relative to the high-light plane have a significant effect on the spot size. Similar to the performance measurements of Olson Edwin for AprilTag [12], we measured and demonstrated the experimental performance in terms of the vertical distance from the subject to the high-light plane and the angle between the subject direction and the normal vector of the high-light plane, respectively.

We take the average of the prediction results of valid test data collected from region $R(i,j)$ under pose p as the prediction result of $Pred(i,j,p)$

$$Pred(i,j,p) = \{Pred(i,j,p)_L, Pred(i,j,p)_P\},$$

where

$$Pred(i,j,p) = \frac{\sum_{n=0}^{60} Est(i,j,p)_n}{60},$$

$$Est(i,j,p)_n = \frac{\sum_{k=0}^{m} e(i,j,p)_k}{m},$$

where m is the number of valid video frames of the n^{th} video of test data acquired at $R(i,j)$ and $e(i,j) = \{e(i,j,p)_L, e(i,j,p)_P\}$ is the prediction result of video frames.

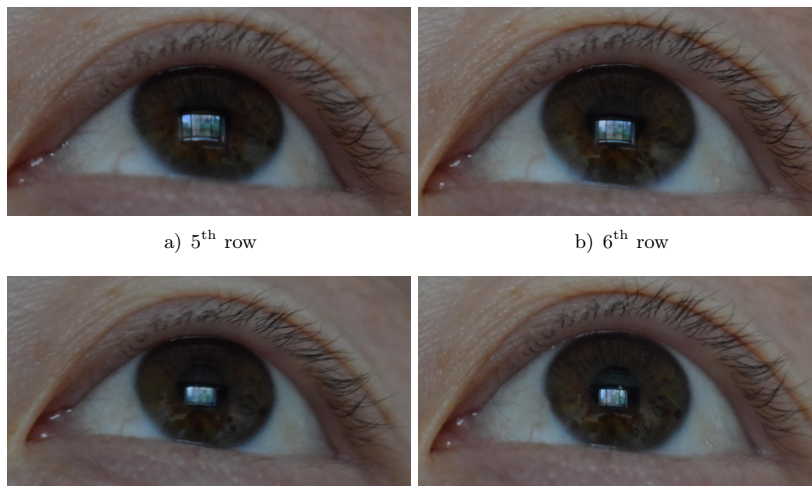
The localization offset range of region $R(i,j)$ under pose p can be decoupled into position estimation offset ranges and pose estimation offset ranges. We keep the pose of the subject equal to $0°$ when measuring the position estimation performance, while we analyze the pose prediction data collected from a fixed region when measuring the pose estimation performance.

We count the average range of the position localization offset ranges for all regions with distance i to the high-light plane as $LOffset(i)$.

$$LOffset(i) = \frac{\sum_0^8 |GT(i,j,0)_L - Pred(i,j,0)_L|}{8},$$

where $GT(i,j,p)_L$ is the location ground truth under the pose p in the $R(i,j)$. To illustrate the ability of our method to cope with different high-lighting environments, we have calculated the position localization offset ranges in L_1, L_2 and L_3 highlighting environment, respectively.

Figure 9 shows the relationship between i and $LOffset(i)$. It can be seen that the value of $LOffset(i)$ increases with i in the three different highlighting areas. As show in Figure 8, we found that the area of the corneal reflection spot decreased rapidly with increasing distance between the subject and the high light plane. When the subject was in row 7, the area of the spot was too small and insensitive to changes in position and posture. As a result, the prediction offset ranges located here becomes larger, the standard deviation of the predicted data increases, and the dispersion of the data becomes more pronounced.



a) 5th row b) 6th row

c) 7th row d) 8th row

Figure 8. The eye images are shot while the subject is standing in rows 5, 6, 7 and 8. By comparison, it can be found that the spot area in the eye image in rows 5 and 6 has a clear shape change, compared to the spot in rows 7 and 8, where the shape change is not obvious enough.

As shown in Figure 7, the proportion of valid data is highest at $R(4,4)$, which means that the performance of pose estimation is more representative than other regions. We convert the pose from the Euler angle to the orientation vector $\boldsymbol{v}$,

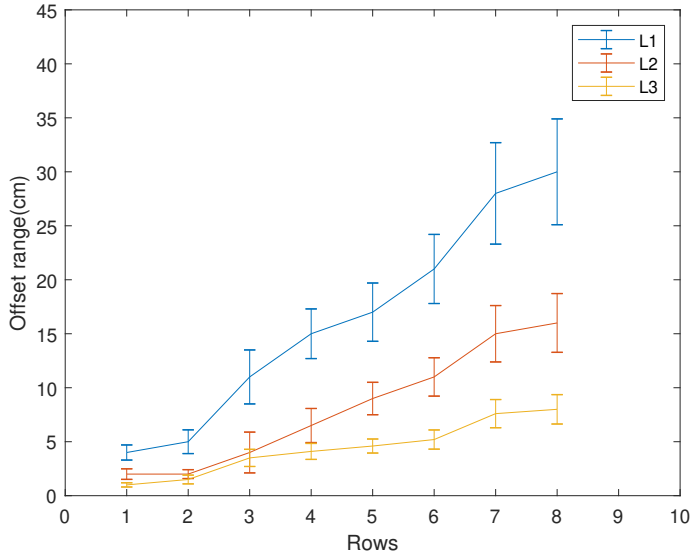


Figure 9. The average location positioning offset ranges increases when the subject is far away from the high-light plane

then we can get the angle between the normal vector of the high-light plane and $\boldsymbol{v}$ is

$$\{0°, 15°, 30°, 33.22°, 41.4°, 45°, 52.2°, 60°, 64.3°\}.$$

We count the pose estimation offset ranges $POffset(p)$

$$POffset(p) = |GT(4,4,p)_P - Pred(4,4,p)_P|,$$

where $GT(i,j,p)_P$ is the pose ground truth in $R(i,j)$. The performance is shown in Figure 10.

We still examined the performance in L_1, L_2 and L_3 high-light scenes, respectively. Similar to the offset ranges distribution of the position estimation, the offset range increases with the orientation angle. However, the incremental gradient is much smaller than the gradient of the position estimation. After examining the original data and the prediction results, we found that at larger orientations, some subjects' eye images were not complete enough, which could easily lead to significant offset ranges in predictions.

5 DISCUSSION

The privacy disclosure risk existing in the social network sites has become a hot topic of concern in recent years. In this paper, we analyze the corneal imaging

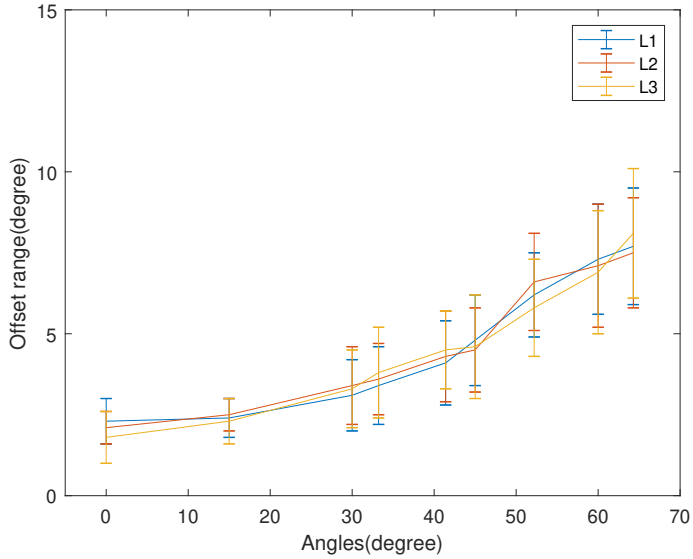


Figure 10. The estimated offset ranges of the postural in the region $R(4, 4)$ indicated that the offset range grows slowly as the pose of the subjects increased

process in a selfie scenario, and design a validation experiment to the subjects with different heights in different indoor scenes. Experiments show that selfies and videos taken in high-light scenes on social media do carry some risk of privacy disclosure for location, posture, height, and lifestyle habits. Our work contributes to the privacy protection of v-loggers and also provides theoretical support for privacy-sensitive groups.

Every coin has two sides. Although the CI can cause the privacy disclosure trouble to selfie taker, the CIS based localization method can be applied in other fields, like disaster rescue, crime tracking, human-machine interaction and computer graphics modeling. In addition, thanks to the weak quality of corneal reflection imaging, a balance of privacy protection and convenience can be achieved when CIS based positioning methods are used in daily life.

Our research is innovative and enriches the research and application scenarios of the CIS. It has been experimentally demonstrated that corneal reflection can reveal the approximate pose and position of the photographer in the room. With the development of social media, there are a large number of selfie videos and photos on Youtube, Twitter and TikTok, including a large number of photos and videos of faces facing windows, screens, etc. These images and videos contain corneal reflection areas that contain information about the subject's pose, which can be used to improve human-machine interaction and expand the application scenarios of human-machine interaction under the constraints of the law. The corneal images

can be shot easily with non-aggressive to others, and as wearable hardware devices evolve, CIS-based positioning efforts can be senselessly integrated into existing smart devices, such as smart glasses, as well as AR and VR devices while being non-intrusive to personal space.

From the above experimental results, we can see that the performance of the CI in the indoor position and posture estimation is related to the area of the high-light and the position of the distance from the high-light. We depend on the solid angle to evaluate the influence of these two aspects.

The solid angle Ω_c presents the reflectable range of the cornea. Let the angle between the normal vector $\boldsymbol{n}_{(0,0)}$ of the corneal curvature vertex O_c and the direction of the optical axis of the camera be σ, Ω_c decreases with increasing σ. For any reflected infinitesimal space δA, the solid angle is

$$\Omega_A = \frac{\delta A \cos^3 \phi(i,j)}{z^2},$$

where $\phi(i,j)$ is the angle between $\boldsymbol{n}_{(0,0)}$ and the normal $\boldsymbol{n}_A$ of δA. Then the solid angle of the corneal reflectable scene C and the high-light L are

$$\Omega_C = \sum \frac{\delta C \cos^3 \phi(m,n)}{z^2},$$

$$\Omega_L = \sum \frac{\delta L \cos^3 \phi(i,j)}{z^2},$$

where $i \in [0,m]$, $j \in [0,n]$, then the scale of the bright pattern in the CI can be expressed as

$$p = \Omega_L / \Omega_C. \tag{5}$$

It is known that Ω_C is constant when σ is constant, and for the sake of calculation, it is assumed that $\delta L = 1$. In general, since the $\Omega_C > 2\pi$ of the cornea, the FOV of the corneal reflection is larger than that of the hemispherical reflective surface. However, due to the influence of the corneal geometry, the resolution at the edge of the cornea decreases sharply compared to that above the pupil. The projected image of the scene at the edge of the cornea piles up and compresses [18]. Therefore, in this paper, when measuring the proportion of the bright pattern in the CI by Equation (5), the value of Ω_C should be less than 2π, and the ratio of the area of the light pattern formed by the projection of the high-light on the cornea to the area of the whole cornea reflectable range is shown in Figure 11.

L_1, L_2 and L_3 are high-light scenes with different sizes that have been used in the experimental section. The size of the bright pattern in the CI changes slowly with increasing Z-axis compared to the position near the high-lights. Also, regions with the same Z value have different localization accuracy. When the distance d between the subject s and O_w increases, the rate of change of Ω_L decreases and p is too similar to be distinguished from nearby regions.

Our work can estimate the posture and position of the subject without additional devices. However, our method can only estimate the posture of head instead of

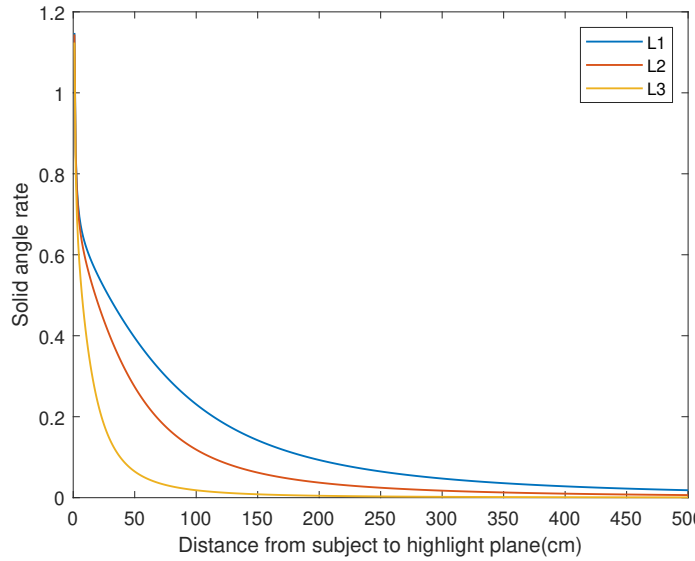


Figure 11. The effect of the subject position to the size of the bright pattern in CI

the whole body. Moreover, from the experimental results, our method cannot be applied to fine body motion tracking applications. Besides, the accuracy of pose and position estimation can hardly be improved without the assistance of the additional equipment.

6 CONCLUSION

In this paper, we demonstrate theoretically and experimentally that selfie images contain private information such as the position and pose of the selfie taker. Our work can help users of social networks to protect their privacy further. In addition, indoor localization by CI is promising in the field of low-accuracy indoor localization, which is less intrusive and less equipment friendly than existing methods. In our future work, we will focus on how to improve the accuracy of CIS for indoor positioning and try to further develop and expand its application.

REFERENCES

[1] Lowe, D. G.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, Vol. 60, 2004, No. 2, pp. 91–110, doi: 10.1023/B:VISI.0000029664.99615.94.

[2] Harris, C.—Stephens, M.: A Combined Corner and Edge Detector. Proceedings of the Fourth Alvey Vision Conference (AVC 1988), Alvey Vision Club, 1988, pp. 147–151, doi: 10.5244/C.2.23.

[3] Hartley, R.—Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, 2003, doi: 10.1017/CBO9780511811685.

[4] Feiner, S.—MacIntyre, B.—Seligmann, D.: Knowledge-Based Augmented Reality. Communications of the ACM, Vol. 36, 1993, No. 7, pp. 53–62, doi: 10.1145/159544.159587.

[5] Chen, J.—Granier, X.—Lin, N.—Peng, Q.: On-Line Visualization of Underground Structures Using Context Features. Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology (VRST '10), 2010, pp. 167–170, doi: 10.1145/1889863.1889898.

[6] Mur-Artal, R.—Montiel, J. M. M.—Tardos, J. D.: ORB-SLAM: A Versatile and Accurate Monocular SLAM System. IEEE Transactions on Robotics, Vol. 31, 2015, No. 5, pp. 1147–1163, doi: 10.1109/TRO.2015.2463671.

[7] Wang, Z.—Zhang, J.—Chen, S.—Yuan, C.—Zhang, J.—Zhang, J.: Robust High Accuracy Visual-Inertial-Laser SLAM System. 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019, pp. 6636–6641, doi: 10.1109/IROS40897.2019.8967702.

[8] Zhang, J.—Gui, M.—Wang, Q.—Liu, R.—Xu, J.—Chen, S.: Hierarchical Topic Model Based Object Association for Semantic SLAM. IEEE Transactions on Visualization and Computer Graphics, Vol. 25, 2019, No. 11, pp. 3052–3062, doi: 10.1109/TVCG.2019.2932216.

[9] Snavely, N.—Seitz, S. M.—Szeliski, R.: Photo Tourism: Exploring Photo Collections in 3D. ACM Transactions on Graphics, Vol. 25, 2006, No. 3, pp. 835–846, doi: 10.1145/1141911.1141964.

[10] Wu, C.: Towards Linear-Time Incremental Structure from Motion. 2013 International Conference on 3D Vision – 3DV 2013, IEEE, 2013, pp. 127–134, doi: 10.1109/3DV.2013.25.

[11] Fischler, M. A.—Bolles, R. C.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Communications of the ACM, Vol. 24, 1981, No. 6, pp. 381–395, doi: 10.1145/358669.358692.

[12] Olson, E.: AprilTag: A Robust and Flexible Visual Fiducial System. 2011 IEEE International Conference on Robotics and Automation, 2011, pp. 3400–3407, doi: 10.1109/ICRA.2011.5979561.

[13] Kato, H.—Billinghurst, M.: Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System. Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR '99), 1999, pp. 85–94, doi: 10.1109/IWAR.1999.803809.

[14] Tsumura, N.—Dang, M. N.—Makino, T.—Miyake, Y.: Estimating the Directions to Light Sources Using Images of Eye for Reconstructing 3D Human Face. Color and Imaging Conference, Vol. 11, 2003, No. 1, pp. 77–81, doi: 10.2352/CIC.2003.11.1.art00014.

[15] Nishino, K.—Nayar, S. K.: Eyes for Relighting. ACM Transactions on Graphics (TOG), Vol. 23, 2004, No. 3, pp. 704–711, doi: 10.1145/1015706.1015783.

[16] Wang, H.—Lin, S.—Liu, X.—Kang, S. B.: Separating Reflections in Human Iris Images for Illumination Estimation. Tenth IEEE International Conference on Computer Vision (ICCV '05) Volume 1, Vol. 2, 2005, pp. 1691–1698, doi: 10.1109/ICCV.2005.215.

[17] Nitschke, C.—Nakazawa, A.: Super-Resolution from Corneal Images. Proceedings of the British Machine Vision Conference (BMVC 2012), BMVA Press, 2012, `https://bmva-archive.org.uk/bmvc/2012/BMVC/paper022/index.html`.

[18] Nishino, K.—Nayar, S. K.: Corneal Imaging System: Environment from Eyes. International Journal of Computer Vision, Vol. 70, 2006, No. 1, pp. 23–40, doi: 10.1007/s11263-006-6274-9.

[19] Swaminathan, R.—Grossberg, M. D.—Nayar, S. K.: Caustics of Catadioptric Cameras. Proceedings Eighth IEEE International Conference on Computer Vision (ICCV 2001), Vol. 2, 2001, pp. 2–9, doi: 10.1109/ICCV.2001.937581.

[20] Nitschke, C.—Nakazawa, A.—Takemura, H.: Eye Reflection Analysis and Application to Display-Camera Calibration. 2009 16$^{\text{th}}$ IEEE International Conference on Image Processing (ICIP), 2009, pp. 3449–3452, doi: 10.1109/ICIP.2009.5413852.

[21] Suda, S.—Yamagishi, K.—Takemura, K.: User Calibration-Free Method Using Corneal Surface Image for Eye Tracking. Proceedings of the 12$^{\text{th}}$ International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2017) – Volume 6: VISAPP, SciTePress, 2017, pp. 67–73, doi: 10.5220/0006100100670073.

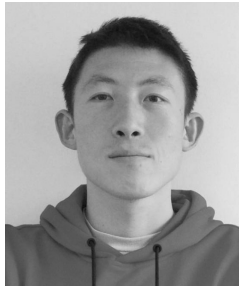
[22] Nakazawa, A.—Kato, H.—Nitschke, C.—Nishida, T.: Eye Gaze Tracking Using Corneal Imaging and Active Illumination Devices. Advanced Robotics, Vol. 31, 2017, No. 8, pp. 413–427, doi: 10.1080/01691864.2016.1277552.

[23] Lander, C.—Löchtefeld, M.—Krüger, A.: hEYEbrid: A Hybrid Approach for Mobile Calibration-Free Gaze Estimation. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, Vol. 1, 2018, No. 4, Art. No. 149, doi: 10.1145/3161166.

[24] Ohshima, Y.—Maeda, K.—Edamoto, Y.—Nakazawa, A.: Visual Place Recognition from Eye Reflection. IEEE Access, Vol. 9, 2021, pp. 57364–57371, doi: 10.1109/ACCESS.2021.3071406.

[25] Du, M.—Chen, K.—Zhang, J.—Liu, H.: A Swift Gaze Estimate Method Based on the Corneal Image System. 2022 IEEE 25$^{\text{th}}$ International Conference on Computer Supported Cooperative Work in Design (CSCWD), 2022, pp. 734–739, doi: 10.1109/CSCWD54268.2022.9776291.

[26] Kaufman, P. L.—Alm, A.: Adler's Physiology of the Eye: Clinical Applications. Mosby, 2003.



Mengqi Du received his Master's degree in computer science and technology from the Zhejiang University of Technology in 2018. He is currently pursuing his Ph.D. degree in the Department of Computer science, Zhejiang University of Technology. His research interests include computer graphics, computer vision, human attention analysis and human-computer interaction.

Yue Zhang received his Master's degree in computer science and technology from the Zhejiang University of Technology in 2017. He is currently pursuing his Ph.D. degree in the Department of Computer science, Zhejiang University of Technology. His research interests include intelligent system, biological signal processing, human-computer interaction and adaptive learning methods.

Jianhua Zhang received his Ph.D. from the University of Hamburg, Hamburg, Germany in 2012. He is currently Professor with the School of Computer Science and Engineering, Tianjin University of Technology, Tianjin, China. His current research interests include SLAM, 3D vision, reinforcement learning, and machine vision.

Honghai Liu received his Ph.D. degree in robotics from the Kings College London, London, U.K., in 2003. He is the Chair Professor of Human Machine Systems, University of Portsmouth, Portsmouth, U.K. His research interests include biomechatronics, pattern recognition, intelligent video analytics, intelligent robotics, and their practical applications with an emphasis on approaches that could make contribution to the intelligent connection of perception to action using contextual information.

Computing and Informatics, Vol. 42, 2023, 1233–1254, doi: 10.31577/cai_2023_5_1233

PREDICTION OF STRESS LEVEL FROM SPEECH – FROM DATABASE TO REGRESSOR

Marián Trnka, Sakhia Darjaa, Róbert Sabo, Milan Rusko*

Institute of Informatics, Slovak Academy of Sciences
Bratislava, Slovakia
e-mail: {trnka, utrrsach, robert.sabo, milan.rusko}@savba.sk

Meilin Schaper, Tim Stelkens-Kobsch

Institute of Flight Guidance, German Aerospace Center
Braunschweig, Germany
e-mail: {meilin.schaper, tim.stelkens-kobsch}@dlr.de

Abstract. The term stress can designate a number of situations and affective reactions. This work focuses on the immediate stress reaction caused by, for example, threat, danger, fear, or great concern. Could measuring stress from speech be a viable fast and non-invasive method? The article describes the development of a system predicting stress from voice – from the creation of the database, and preparation of the training data to the design and tests of the regressor. StressDat, an acted database of speech under stress in Slovak, was designed. After publishing the methodology during its development in [1], this work describes the final form, annotation, and basic acoustic analyses of the data. The utterances presenting various stress-inducing scenarios were acted at three intended stress levels. The annotators used a "stress thermometer" to rate the perceived stress in the utterance on a scale from 0 to 100. Thus, data with a resolution suitable for training the regressor was obtained. Several regressors were trained, tested and compared. On the test-set, the stress estimation works well (R square = 0.72, Concordance Correlation Coefficient = 0.83) but practical application will require much larger volumes of specific training data. StressDat was made publicly available.

Keywords: Acoustic correlates of stress, speech under stress, stress database, stress-inducing scenarios, stress measurement

* Corresponding author

1 INTRODUCTION

As soon as in 1983, Streeter et al. noted that: "Voice indications of psychological stress are perhaps the most commonly studied emotional phenomena in speech production." [2] The interest of scientists in this topic still continues, and the development of technologies based on deep neural networks makes it possible to project theoretical knowledge into applications. The level of stress in a human being corresponds to the levels of stress hormones, for which fast non-invasive online measurement is not yet available. Therefore, various bio-signals are used to estimate stress levels (for a comprehensive survey see [3]). It is assumed that acoustic symptoms contained in speech can be used to identify increased levels of momentary (acute) stress [4, 5].

One of the major obstacles to this research is the lack of representative databases of speech under stress, with reliable assessment of stress levels. Creating a database of recordings of real-life stress-inducing situations and reliably assessing the level of current stress is extremely challenging, as ethical and health reasons do not allow scientists to expose subjects to high levels of stress. However, several attempts have been made to create databases of speech under stress, either acted or induced (e.g., [4]).

Defining "stress" is a notoriously difficult problem. No single definition will satisfy all circumstances, or, if it does, it will be too vague to have any practical use. Our definition was chosen so that it is appropriate for certain security applications of speech technology but may be unsuited to other areas, such as medical research. The issues of stress definition and measurement were well summarized in detail in the work of Epel et al. [6] and a unified view of stress was proposed.

In contrast to chronic stress, our present work focuses on the acute, or momentary stress that is experienced as an immediate perceived threat, either physical, emotional, or psychological [7]. Epel notes that "Acute psychological stress responses are often measured by capturing specific emotional states. This is because negative emotional responses (fear, anxiety, sadness, anger) to an acute stressor are considered a core component of an acute stress response." [6]. In their previous work, the authors of this paper also dealt with the possibility of measuring stress via the identification of expressed emotions, as emotions with high arousal and negative valence can be triggered by immediate stress [8]. However, the problem with this approach was that they were not able to estimate valence reliably, although this applied primarily to emotional expressions with lower emotional arousal. Therefore, in this work, we do not identify emotions, but instead, we try to estimate stress (i.e., the annotators were instructed to rate the perceived level of stress). Our aim is to design an automatic system that monitors the level of stress from acoustic cues in speech. Their increased intensity could indicate a worsened, dangerous, or critical situation. Speech is a psychophysiological process, influenced by environmental and/or internal challenges and reflects the level of stress [9], which entitles us to believe that the acoustic properties of speech can be used to estimate stress. As an example of an attempt at practical application, we can mention the Stressometer

and Stress Tracker – a mobile application that provides the user with a graphic representation reflecting the measured stress [10].

2 MATERIALS AND METHODS

Due to the lack of databases of speech under stress, representing critical situations, thoroughly annotated, and large enough for training, we decided to record our own database. To have control over the content of recordings (and not being able to record in real critical situations) we have chosen to record an acted speech database. We believe that the manifestations of stress in various stressful situations played by actors are, in the first approximation, sufficiently like spontaneous ones. So, they can be used for research of acoustic cues of speech under stress and for training and testing automatic systems for stress measurement.

In our former research, we experimented with training a stress-predictor [11] on an acted database CRISIS [12], containing utterances with various levels of emotional arousal (calming, neutral, warning and highly insisting). This database was originally designed for expressive speech synthesis purposes. Using it for stress-detector training was only a make-do solution. However, the functionality of the stress prediction was acceptable, and the stress-detector was implemented as an add-on tool to increase the security of air-traffic control [13].

2.1 StressDat Database

A new, more specialized database of speech under stress had to be designed. As the authors are primarily engaged in research and development of applications for the Slovak language, the database is in Slovak. The methodology and details of the content of this database have already been published in [1] as a work under progress. The actual paper briefly summarizes its final form and properties and presents how it was used for the development of regressors for predicting the level of stress from speech.

2.1.1 Database Creation Method

The database creation method assumes that appropriately designed sets of sentences corresponding to stress-inducing scenarios can be played with high naturalness at different stress intensities. Twelve scenarios of the stress-inducing situations were drawn up. Each scenario consists of 10 to 13 sentences with semantic content and a form that can be played at three levels of stress (neutral, low, high). In addition, four emotionally neutral situations with sentences corresponding semantically to the particular neutral (non-stress) situation were prepared to reach "really neutral speech", uninfluenced by the semantic content of the stress-inducing scenarios.

For the purpose of training a regressor, it is necessary to have utterances with an assigned stress value available. The values should cover as large a range as possible and should be able to reach all the values of this interval. If analog representation

is not possible, the resolution of the digital representation should be as high as possible. However, from the previous recording of expressive speech databases, we had the experience that speakers were not able to consistently maintain more than three levels of expressiveness when increasing from neutral speech through expressive to extremely expressive. We therefore decided to instruct the actors to play three levels of stress – neutral speech, speech under stress and speech under extremely high stress. We will refer to these levels as three intended levels. We could accept a number representing the level of stress that the actor tried to express during the realization of the given utterance (i.e., 1, 2, or 3) as one way of evaluating the intensity of stress. But the resolution of three levels only is pitifully small. However, each person has a different range of stress manifestation intensity in speech, which means that many speakers will jointly cover a larger range than an individual speaker. On the other hand, the intensity of stress cues in utterances with different intended levels of different speakers (or even of the same speaker) can be perceived as similar by the listener. The boundaries of intended levels clusters overlap on the receiving part of the communication. Therefore, we decided to use the rating of stress, perceived in the utterance by annotators. The interface for collecting ratings, which uses a so-called stress-thermometer will be described later in Section 2.1.3.

2.1.2 Recorded Subjects

Our subjects were professional and non-professional actors recruited mostly from the pool of personal contacts of the authors. The database was created at a time of COVID-19 pandemic, so it was not possible to record in studio. The actors recorded their speech using their own phones in their own apartments. They received detailed instructions for positioning the smartphones and selecting a suitable acoustic environment to ensure as similar recording conditions across the speakers as possible [1]. Recordings were captured using the default voice-recorder application of a specific actor's phone with the highest available audio quality and stored in the .wav or .m4a format. All audio files were later down-sampled to 16 kHz, 16-bit, mono. The final database includes 30 speakers (16 females, 14 males). The subjects received a small payment for recording. The average age of the speakers was 29 years. The youngest was 21 and the oldest was 38. During the recording of the database, we came to the decision, that the length of the audio material from each recorded speaker was too small, and that the "neutral" level can be influenced by the semantic content of the sentences of the stress-inducing scenarios. 6 more scenarios were therefore added (3 semantically neutral situations to be presented at "really neutral" level, and 3 additional stress-inducing situations to be presented at three levels of stress). 20 speakers recorded the full battery of 16 situations and 10 speakers recorded 10 situations in 3 levels (situations No. 1, 2, 3, 5, 6, 8, 9, 10, 11, 12) and 2 neutral situations (situations 13, 14). The detailed description of the scenarios is presented in Table 1. The transcription of each recording was manually corrected. So, the text representation of each recorded utterance is exact even in case the actor made a mistake, pronounced a different word, or omitted a word from the scenario. In the

following, we will refer to the intended stress levels as a – neutral, b – medium, and c – high.

Category	**No.**	**Intended Stress Level**	**Description**
Threat of losing control over the situation	1	a, b, c	As an airline pilot you need to make an emergency landing.
	2	a, b, c	Navigating a plane at the airport during very bad weather.
	3	a, b, c	As a pilot you need an undisciplined passenger to comply with the ban on using laptops during take-off/landing.
	4	a, b, c	As a firefighter coordinator you organize firefighting in a burning building.
Psycho-social stress	5	a, b, c	As a parent, you must organize the morning routine for your kids before leaving for school.
	6	a, b, c	You are finishing last-minute changes for an important presentation with a colleague.
	7	a, b, c	As a passenger, you need information on train departures urgently.
The threat of life/health injury of self/close ones	8	a, b, c	You call an ambulance for your father who suffered a stroke.
	9	a, b, c	You are trying to pacify your drunk brother who is trying to forcefully enter your flat.
	10	a, b, c	You are calling the police to resolve the situation with your drunken brother above.
	11	a, b, c	As a pilot, you organize evacuation from a burning aircraft.
	12	a, b, c	You report an insured event after a car accident by phone.
Neutral	13	a	You talk about school with your son.
	14	a	You are buying shoes.
	15	a	You teach students at school.
	16	a	You are reading a text to a colleague.

Table 1. Description of the scenarios

2.1.3 Annotation of the Perceived Stress Level by Listeners

The perceived stress level in the recorded sentences was rated by five annotators. We have created a simple graphic interface for utterances evaluation. The annotators evaluated the utterance in sets of 100 with shuffled intended stress levels. They were advised to take breaks of at least 5 minutes between sets. As the task demands high attention, they were strongly advised not to evaluate more than 3 sets per day, which

corresponds to approximately 1 hour of work. In each set, there were 100 utterances, but each annotator had a different order of sentences in the corresponding set in order to minimize the influence of the previously heard sentences on the evaluation. To limit the speaker influence, the annotators evaluated sentences from several different speakers in each set. Annotators were instructed to assess "distress, anxiety or discomfort on a scale of 0 to 100". They were told: "Imagine you have a thermometer that measures the stress according to the indicated scale. Rate on that scale how you think the person felt when pronouncing the utterance."

The descriptions for the individual points of the scale were the following (translated from Slovak):

0: Totally relaxed;

10: Alert, well concentrated;

20: Minimal anxiety/discomfort;

30: Mild anxiety/discomfort, does not interfere with performance;

40: A little upset, manages the activity but does not feel well;

50: Moderate anxiety/discomfort, feels uncomfortable but can continue activities;

60: Moderate to severe feeling of anxiety/discomfort;

70: Quite anxious/significant discomfort, it disrupts activity;

80: Great anxiety/discomfort, unable to concentrate;

90: Extreme anxiety/discomfort;

100: The most intense anxiety/fear/discomfort he/she has ever felt.

A picture of a thermometer with a verbal description of the states corresponding to individual "temperature" degrees is often used as a good aid in the more detailed evaluation of affective phenomena on a wide scale. In general, the scale can be continuous, but the finite number of "degrees" leads to discretization (e.g. [14, 15, 16, 17, 18]). We used a "Stress Thermometer" tool (see Figure 1) based on the Subjective Units of Distress Scale [19].

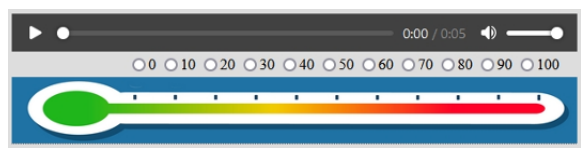


Figure 1. "Stress Thermometer" tool that allows the annotator to listen to the utterance and to assign a perceived stress level

2.1.4 Inter-Annotator Agreement

In [1] the authors assessed the agreement among the annotators using Fleiss' kappa reaching values around 0.3, which suggests a fair agreement. The interrater correlation coefficient (ICC), however, provides a better estimate of the agreement among raters [20] since the distance between the disagreements on the scale is taken into consideration (those between adjacent values are considered better than disagreements spanning multiple scale levels). We thus created a 2-way model based on a mean-rating ($k = 5$) and absolute agreement within the irr package [21] in R, which returned $\mathrm{ICC}(A, 5) = 0.935$ and 95 % confident intervals $(0.921, 0.945)$ and a more conservative model based on single rater gives $\mathrm{ICC}(A, 1) = 0.74$. These values correspond to excellent and moderate-good reliability, respectively.

To account for the inter-annotator variability, z-score normalization was used and linearly projected on the 0–100 interval. All the following calculations were performed on this normalized annotation.

In accordance with the methodology, all annotators were Slovak native speakers. However, we also tried to evaluate the utterances by two annotators who do not speak, nor understand Slovak at all. One was from India (IND) and one was from the Netherlands (NL). Both live in Europe in an English-speaking environment. While NL's evaluations were in good agreement with Slovak annotators, IND used only a very narrow range of values. He said he was not able to perceive well the presence and intensity of stress in Slovak speech. Although it is possible that this was due to the individuality of the annotators, it may also indicate that the evaluation is strongly culturally dependent. Figure 2 shows how the three acted (intended) stress levels were subjectively rated on a scale from 0 to 100 by the annotators (perceived stress levels).

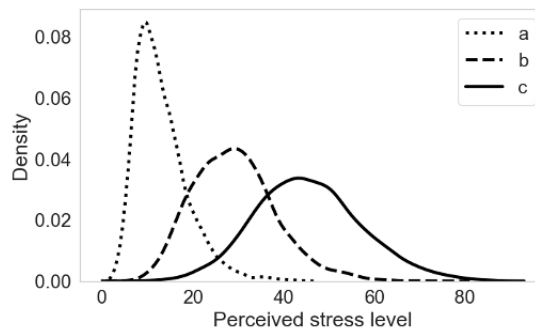


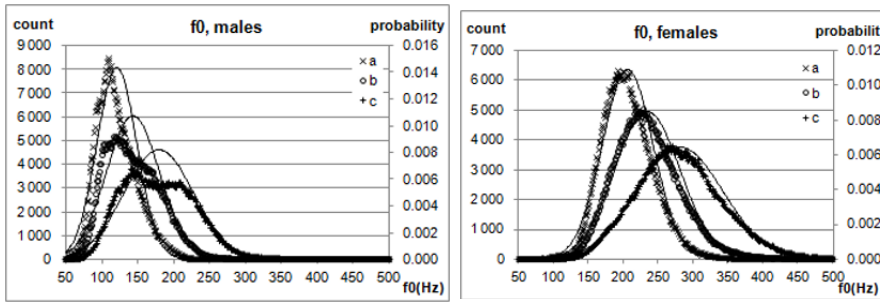
Figure 2. Distribution of perceived levels assigned by the annotators to the utterances with respect to the three intended stress levels

2.2 Acoustical Analyses

To get an idea of how the acoustic properties of speech differ at individual levels of stress in StressDat, analyzes of characteristics representing prosody and voice quality were performed, namely fundamental frequency of vocal fold vibrations, sound pressure level, long-term-average-spectrum, formant positions of vowels, speech rate and counts and durations of pauses.

2.2.1 Fundamental Frequency (F0)

F0 was measured on 25 ms frames through the entire corpus. The distribution of F0 values and their Gaussian approximation curves for the three intended levels of stress in male and female speakers are presented in Figure 3 a) and 3 b), respectively.



a) b)

Figure 3. Histograms of F0 values with Gaussian approximation curves for a) male and b) female speakers and three levels of intended stress (a – neutral, b – medium, and c – high)

The values of F0 increase with increasing stress level. While the distributions of female speakers are well approximated with Gaussians, the distribution of F0 in male utterances of levels b and c looks more like mixtures of two Gaussians. We can only speculate that some parts of these utterances are pronounced with moderate voice effort and the other parts are more expressive. This depends on the actor's way of realization (and thus also on the choice of the actors).

Mean F0 was computed per speaker and stress level. The average value of the F0 means and standard deviations with respect to the three levels of intended stress are presented in Table 2.

2.2.2 Sound Pressure Level

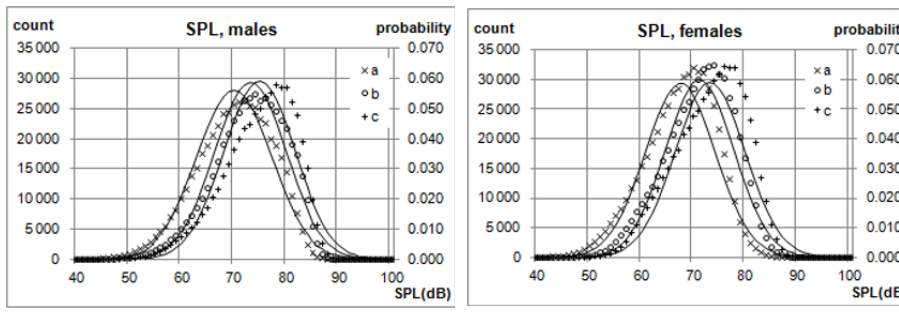
Sound pressure level of the speech signal is known to be highly correlated with manifestations of affect in speech. According to Praat documentation, "Intensity object represents an intensity contour at linearly spaced time points $t_i = t_1 + (i -$

Intended Stress Level	Average F0 Mean [Hz]		Average F0 stdev [Hz]	
	Males	Females	Males	Females
a	120.9	205.8	27.8	36.6
b	143.4	234.1	37.1	47.1
c	179.4	281.0	48.7	62.0

Table 2. Average F0 mean and standard deviation

1) dt, with values in dB SPL, i.e. dB relative to $2{\cdot}10^{-5}$ Pascal, which is the normative auditory threshold for a 1 000 Hz sine wave" [22].

Correct interpretation of the correlation between intensity and stress is hampered by the fact that differences in SPL (which reflect vocal effort [23]), are highly non-specific. They are influenced not only by stress, but also by the distance of the speaker from the addressee, directional orientation, Lombard effect, emotion, mood, and personality of the speaker. In addition, a calibrated measuring system would be needed to measure SPL correctly [24]. Phone recording presents the problem of automatic gain control, which is specific to each phone model and its parameters are unknown to the user. The following analyses are therefore only informative, valid for this database, and cannot be freely generalized. The histograms of SPL values with Gaussian approximation curves for a) males, and b) females, at three levels of intended stress are presented in Figure 4.



a) b)

Figure 4. Histograms of SPL values with Gaussian approximation curves for a) males, b) females, and three levels of intended stress (a – neutral, b – medium, and c – high)

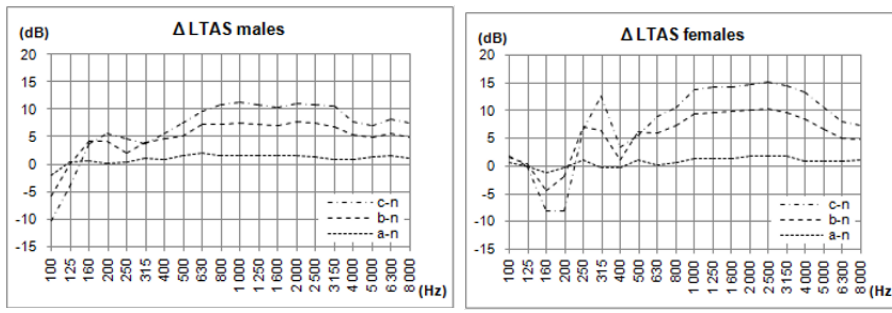
Female speakers produced speech with a slightly lower SPL, but otherwise the histograms of male and female subjects do not differ.

2.2.3 Long-Term-Average-Spectrum (LTAS)

Long-term-average-spectrum (LTAS) is known to contain information on the vocal effort [25], which is one of the means used by speakers to express various levels of expressivity.

Rather than measuring the overall amplitude, some studies have tended to determine the amplitude of different frequency bands. Although different techniques were employed to achieve these calculations, there is general agreement that amplitude measurements are greater in higher frequencies i.e., above 1 000 Hz in stress conditions compared to control conditions [26].

As it was already mentioned, before the recording of the three levels of stress (a, b, c), the speakers were asked to relax and record a set of emotionally neutral sentences to get an independent sample of "really neutral" speech (i.e., speech not influenced by stress). This level is referred to as "n" or "level n" in Figure 5. To study the behavior of LTAS at various stress levels, we took the LTAS of level n as a reference and plotted the deviations of the spectra of levels a, b, c, from this reference value to the graph (Figure 5). This difference spectrum is marked as ΔLTAS. It was analyzed in third-octave frequency bands.



a) b)

Figure 5. Comparison of ΔLTAS for a) male and b) female speakers for 3 intended levels

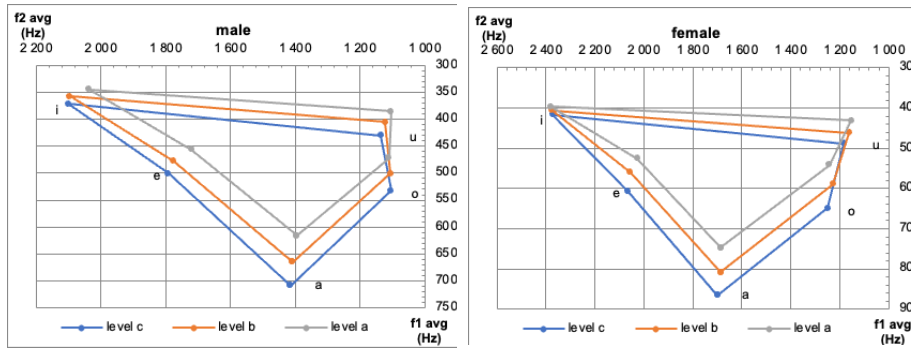
It can be seen from the Figure 5, that biggest differences of LTAS are in the range from 125 to 400 Hz, which is caused by energetically rich F0, shifting higher with increasing stress level. Much more notable are the higher values in the range from 1 000 to 3 150 (males) or to 4 000 Hz (females), which is caused by increasing energy of higher harmonics with increasing vocal effort.

2.2.4 Formant Positions of Vowels (F1, F2)

Ruiz et al. [27] analyzed spectral balance frequency in vowels and tentatively suggested that this might be a promising parameter that is sensitive to stress [26].

We measured the mean frequencies of the formants of the Slovak vowels a, e, i, o, u and presented them in the F1 versus F2 formant space in Figure 6.

While increasing the central frequency of F1 formant (caused mostly by the up-down movement of the jaw) is clearly observable in most of the vowels with increasing levels of stress, the movement of F2 (that depends mostly on the back-front movement of the tongue) is negligible.



a) b)

Figure 6. F1 versus F2 formant diagram of the five Slovak vowels a, e, i, o, u for the three intended levels of stress

2.2.5 Speech Rate

The speech rate of each utterance was measured in syllables per second and in words per second and the results are presented in Table 3.

Intended Stress Level	Syllables per Second		Words per Second	
	Males	Females	Males	Females
a	6.137	6.112	2.916	2.912
b	6.452	6.317	3.069	3.010
c	6.640	6.466	3.155	3.079

Table 3. Speech rate: syllables per second and words per second for 3 intended stress levels

As many of the utterances are short, syllables per second is the more suitable a measure to reflect the speech rate in StressDat than words per second. The syllable rate slightly increases with intended stress level and the gender differences are insignificant.

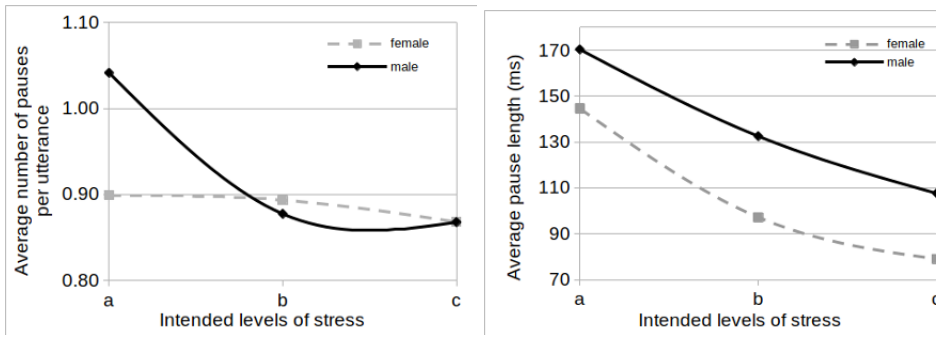
2.2.6 Pause Counts and Durations

Pauses can partly be influenced by respiration, which is the driving force of both stress and voice production, and Van Puyvelde et al. [28] hypothesize it to be the missing link in our understanding of the underlying mechanisms of the dynamic between speech and stress.

All recordings in StressDat were manually transcribed into text and then subjected to forced alignment. This algorithm determined the exact boundaries of phonemes and identified pauses. The average number of pauses per utterance (a)

and mean duration of pauses (b) with respect to three intended levels of stress are presented in Figure 7.

The results for the average number of pauses for men and women are not quite consistent. This characteristic is highly dependent on the individuality of the speaker, and due to the small number of speakers, the statistical "smoothing" of the results was not effective enough. On the other hand, the shortening of average pause lengths with increasing stress levels has the same tendency in men and women. However, pauses for women are approximately 30 ms shorter.



a) b)

Figure 7. Average number of pauses per utterance a) and mean duration of pauses b) in StressDat utterances with respect to three intended levels of stress

2.2.7 Sentence Lengths

StressDat contains sentences of various lengths. The graph of distribution of sentence lengths is shown in Figure 8. The minimum duration of the utterance is 0.6 s, and the maximum is 31 s. The average is 4.57 s. 98.9 % of sentences have a length in the range of 1–10 s.

2.3 Representative Features and Regressors

From a machine learning perspective, the most significant difference between regression versus classification is that while regression helps predict a continuous quantity, classification predicts discrete class labels. However, a regression algorithm can predict a discrete value which is in the form of an integer quantity.

Trnka et al. have already published experiments on the use of three class classificators trained on StressDat [29]. In this work we will focus on the approach using regression.

Experiments with stress level regression focused on comparing 3 diverse types of parameters (GeMAPS, X-vector and TRILL) using 5 diverse types of regres-

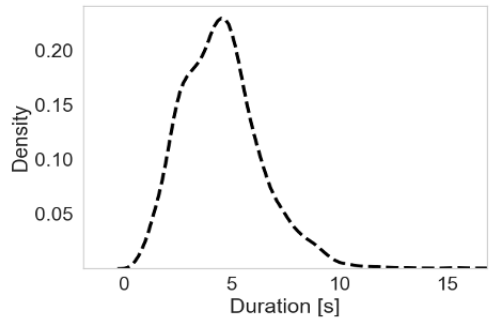


Figure 8. Distribution of the time lengths of the audio files in seconds

sors: Random Forest regressor, Gradient Boosting regressor, Decision Tree regressor, Multi-layer Perceptron regressor and Support Vector Regressor. The default hyperparameters of the regressors were used in training. The Scikit-learn machine learning library [30] was used for training. The training and test sets for regression are organized in pairs of feature vectors (X-vector, TRILL or GeMAPS), representing particular utterances, and the corresponding value of the annotated perceived stress level.

In the data preparation phase, representative characteristics are calculated from the digitized audio signal. We used three different sets of features. Acoustic features were extracted using the OpenSMILE toolkit [31] and the GeMAPSv01b subset of the OpenSMILE features was used. This subset, containing 62 features, was designed especially for affective speech recognition [32]. Therefore, we assumed it will work well for stress prediction.

For comparison we also used modern auditory non-semantic speech representation, X-vectors [33] and TRILLs (TRIpLet Loss network vectors) [34]. We used Kaldi toolkit [35] to train X-vectors extractor to compute the 512-dimensional X-vectors. The procedure was like that presented in [33]. 39 MFCC features (13 MFCC + delta MFCC + delta delta MFCC) were used as the input spectral information for the X-vector extractor. The frame length was 25 ms and frame shift 10 ms. The X-vector extractor was trained on VoxCeleb [36] and VoxCeleb2 [37] speech databases. The energy-based Voice Activity Detector (VAD) was used to filter out silence frames.

To compute TRILL embedding we used publicly available pretrained network [38]. The size of TRILL embedding was 512 – same as X-vector. For TRILL computing we did not use VAD.

3 DATABASE EVALUATION

To evaluate the potential of the created database, it was used to train and test the regressors. We first divided StressDat into independent training set and test set.

As one of the speakers had to be temporarily excluded from the database due to incomplete annotation, the number of speakers was decreased to 29. The train set had 24 speakers (10 female, 12 male) which is 10 591 audio files (utterances) in total. The test set had 5 speakers (2 male, 3 female), which is 2 172 utterances.

Figure 9 shows density of perceived stress levels in the training set and test set.

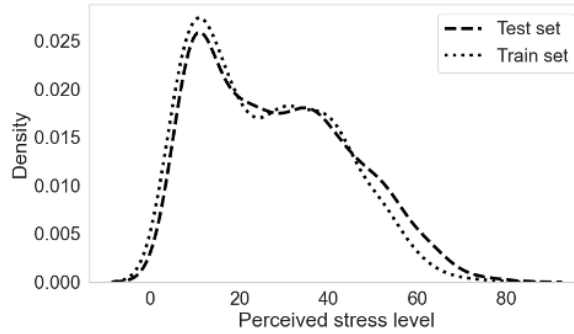


Figure 9. Distribution of perceived stress levels in training set and test set

It is clear from Figure 9, that the largest part of utterances was evaluated as expressing lower levels of stress. Elevated levels are still quite common, but there are few manifestations of extremely high stress. However, the distribution of stress levels in the test set replicates well in the training set.

3.1 Regression Results

3.1.1 Evaluation Metrics

In different works, different measures were used for evaluation of the quality of regression. For better comparison we used 4 metrics to evaluate the quality of the regression: R2 (Coefficient of determination), MAE (Mean Absolute Error), CC (Pearson correlation coefficient) a CCC (Congruence Correlation Coefficient). CCC is a correlation measure, which is widely used in affective speech research. It was used for instance in OMG – Emotion Challenge at IEEE World Congress on Computational Intelligence in 2018 [39].

3.1.2 Comparison of Features and Regressors

Five machine learning algorithms were compared when using three different sets of representative features. Regressors were trained on StressDat training set and tested on StressDat test set. Results are presented in Table 4.

It can be seen that the overall best result is achieved by the Gradient Boosting Regressor model trained on X-vector features (CCC 0.83). However, we can conclude

Regressor	**Feature**	**R^2**	**MAE**	**CC**	**CCC**
Support Vector	X-vector	0.71	7.2	**0.86**	0.8
Random Forest	X-vector	0.64	8.23	0.83	0.75
Gradient Boosting	X-vector	**0.72**	**7.1**	**0.86**	**0.83**
Decision Tree	X-vector	0.27	11.16	0.61	0.6
ML Perceptron	X-vector	0.55	9.03	0.77	0.74
Support Vector	GeMAPS	**0.7**	**7.11**	**0.85**	**0.8**
Random Forest	GeMAPS	0.66	7.77	0.83	0.79
Gradient Boosting	GeMAPS	0.62	8.24	0.82	0.79
Decision Tree	GeMAPS	0.26	10.86	0.64	0.63
ML Perceptron	GeMAPS	0.51	9.29	0.78	0.76
Support Vector	TRILL	**0.64**	7.9	**0.83**	0.74
Random Forest	TRILL	0.57	8.87	0.78	0.68
Gradient Boosting	TRILL	**0.64**	**7.73**	0.81	**0.75**
Decision Tree	TRILL	0.05	12.49	0.48	0.48
ML Perceptron	TRILL	0.23	11.6	0.66	0.66

Table 4. Performance of regression of different acoustic features on various machine learning algorithms

that the most stable results were obtained by the Support Vector Regressor for all three types of parameters. In the CC metric it was always the best (for the X-vector it had the same result as Gradient Boosting).

The comparison of different features turned out best for X-vector embedding. However, the Opensmile GeMAPS parameters performed only slightly worse, even though the vector size was only 62 compared to the X-vector size of 512.

On the contrary, Decision Trees clearly gave the worst results for our experiment setup.

4 DISCUSSION

We proposed a methodology for creating an acted database of speech under stress. We have successfully recorded 30 speakers, which gives in total nearly 16 hours of recordings. We proposed an approach of subjective assessment of stress by the annotators using a so-called stress thermometer. This allowed us to obtain a rating on a continuous scale and within a reasonable range of the intensity of acted manifestations of stress.

The speakers had to produce sentences from scenarios at three intended levels of stress. However, the affective setting is individual for each speaker and the levels are not consistent between speakers. When evaluating the perceived level of stress, the annotators did not have information about the intended level of stress and were therefore not influenced by this information. Their assessment was not limited to three levels, but they could use a scale from 0 to 10, which is suitable for creating a regressor assessing stress on a continuous scale.

We made basic acoustic measurements on the database to provide information about the impact of stress on the acoustic parameters of speech. Separate analyses for male and female speakers make it possible to study gender differences. Through experiments with three types of representative characteristics and with five types of regressors, we tried to design the best combination suitable for automatic system for stress level estimation from speech.

Training was done on the training-set and the tests were done on the test-set of the StressDat database. The speakers in the training and testing sets do not overlap. However, it should be noted that the scenario texts in the training and test sets are the same. Ideally, the quality of stress level prediction should be verified on a completely independent, reliably annotated test set, but this was not available.

Due to the pandemic situation, it was not possible to upload the database in the flight simulator. So, it was replaced by an acted database. This results in less naturalness of speech expression. Moreover, the database is too small to representatively cover all aspects of stress in speech. It is an ad-hoc solution that needs to be improved in the future by obtaining authentic speech data from real stressful situations. In the future, different time windows of stress assessment, both longer and shorter than one sentence, can be considered.

We hope the StressDat database partly fills the lack of databases of speech under stress with a reliable assessment of the stress level. Moreover, this Slovak database opens possibilities for cross-lingual and cross-cultural research.

Stress is an overly broad concept, which covers a large number of different situations and reactions to them. It is therefore highly likely that for the creation of practical applications, it will be necessary to build specific stress databases covering adequately the very manifestations of stress that the system is supposed to detect.

Funding: This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 832969. This output reflects the views only of the authors, and the European Union cannot be held responsible for any use which may be made of the information contained therein. For more information on the project, see: `http://satie-h2020.eu/`. The work was also funded by the Slovak Scientific Grant Agency VEGA, project No. 2/0165/21. The work was also funded by the The Slovak Research and Development Agency, project No. APVV-21-0373.

Institutional Review Board Statement: The study was conducted in accordance with the Declaration of Helsinki and approved by the Ethics Committee of the Slovak Academy of Sciences (protocol code 83296 from September 25, 2019).

Informed Consent Statement: Written informed consent for publication was obtained from all subjects involved in the study.

Data Availability Statement: StressDat database is available for non-commercial purposes on request from the corresponding author.

Conflict of Interest: The authors declare no conflict of interest.

REFERENCES

[1] Streeter, L. A.—Macdonald, N. H.—Apple, W.—Krauss, R. M.—Galotti, K. M.: Acoustic and Perceptual Indicators of Emotional Stress. The Journal of the Acoustical Society of America, 1983, Vol. 73, 1997, No. 4, pp. 1354–1360, doi: 10.1121/1.389239.

[2] Sharma, N.—Gedeon, T.: Objective Measures, Sensors and Computational Techniques for Stress Recognition and Classification: A Survey. Computer Methods and Programs in Biomedicine, Vol. 108, 2012, No. 3, pp. 1287–1301, doi: 10.1016/j.cmpb.2012.07.003.

[3] Hansen, J. H. L.—Bou-Ghazale, S. E.: Getting Started with SUSAS: A Speech Under Simulated and Actual Stress Database. Proceedings of 5th European Conference on Speech Communication and Technology (EUROSPEECH 1997), 1997, doi: 10.21437/Eurospeech.1997-494.

[4] Baird, A.—Triantafyllopoulos, A.—Zänkert, S.—Ottl, S.—Christ, L.—Stappen, L.—Konzok, J.—Sturmbauer, S.—Messner, E.-M.—Kudielka, B. M.—Rohleder, N.—Baumeister, H.—Schuller, B. W.: An Evaluation of Speech-Based Recognition of Emotional and Physiological Markers of Stress. Frontiers in Computer Science, Vol. 3, 2021, Art. No. 750284, doi: 10.1016/j.compind.2017.04.005.

[5] Epel, E. S.—Crosswell, A. D.—Mayer, S. E.—Prather, A. A.—Slavich, G. M.—Puterman, E.—Mendes, W. B.: More Than a Feeling: A Unified View of Stress Measurement for Population Science. Frontiers in Neuroendocrinology, Vol. 49, 2018, pp. 146–169, doi: 10.1016/j.yfrne.2018.03.001.

[6] Plarre, K.—Raij, A.—Hossain, S. M.—Ali, A. A.—Nakajima, M.—Al'Absi, M.—Ertin, E.—Kamarck, T.—Kumar, S.—Scott, M.—Siewiorek, D.—Smailagic, A.—Wittmers, L. E.: Continuous Inference of Psychological Stress from Sensory Measurements Collected in the Natural Environment. Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), 2011, pp. 97–108.

[7] Trnka, M.—Darjaa, S.—Ritomský, M.—Sabo, R.—Rusko, M.—Schaper, M.—Stelkens-Kobsch, T.: Mapping Discrete Emotions in the Dimensional Space: An Acoustic Approach. Electronics, Vol. 10, 2021, No. 23, Art. No. 2950, doi: 10.3390/electronics10232950.

[8] Hansen, J. H. L.—Patil, S.: Speech Under Stress: Analysis, Modeling and Recognition. In: Müller, C. (Ed.): Speaker Classification I. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 4343, 2007, pp. 108–137, doi: 10.1007/978-3-540-74200-5_6.

[9] Gaggioli, A.—Pioggia, G.—Tartarisco, G.—Baldus, G.—Ferro, M.—Cipresso, P.—Serino, S.—Popleteev, A.—Gabrielli, S.—Maimone, R.—Riva, G.: A System for Automatic Detection of Momentary Stress in Naturalistic Settings. In: Wiederhold, B. K., Riva, G. (Eds.): Annual Review of Cybertherapy and Telemedicine 2012. IOS Press, Studies in Health Technology and Informatics, Vol. 181, 2012, pp. 182–186, doi: 10.3233/978-1-61499-121-2-182.

[10] Sabo, R.—Rusko, M.—Ridzik, A.—Rajčáni, J.: Stress, Arousal and Stress Detector Trained on Acted Speech Database. In: Ronzhin, A., Potapova, R., Németh, G. (Eds.): Speech and Computer (SPECOM 2016). Springer, Cham, Lecture Notes in Computer Science, Vol. 9811, 2016, pp. 675–682, doi: 10.1007/978-3-319-43958-7_82.

[11] Rusko, M.—Darjaa, S.—Trnka, M.—Cerňak, M.: Expressive Speech Synthesis Database for Emergent Messages and Warnings Generation in Critical Situations. Proceedings of the Language Resources for Public Security Applications Workshop, LREC 2012, Istanbul, Turkey, 2012, pp. 50–53.

[12] Rusko, M.—Trnka, M.—Darjaa, S.—Rajčáni, J.—Finke M.—Stelkens-Kobsch, T.: Enhancing Air Traffic Management Security by Means of Conformance Monitoring and Speech Analysis. In: Klempous, R., Nikodem, J., Baranyi, P. (Eds.): Cognitive Infocommunications, Theory and Applications. Springer, Cham, Topics in Intelligent Engineering and Informatics, Vol. 13, 2019, pp. 177–199, doi: 10.1007/978-3-319-95996-2_9.

[13] Sabo, R.—Beňuš, Š.—Trnka, M.—Ritomský, M.—Rusko, M.—Schaper, M.—Szabo, J.: StressDat – Database of Speech Under Stress in Slovak. Journal of Linguistics/Jazykovedný časopis, Vol. 72, 2021, No. 2, pp. 579–589, doi: 10.2478/jazcas-2021-0053.

[14] Feelings Thermometer. Wisconsin Office of Children's Mental Health. Available online: `https://children.wi.gov/Pages/FeelingsThermometer.aspx`. [Accessed 2022].

[15] Stress Thermometer. Ohio Center for Autism and Low Incidence. Available online: `https://www.ocali.org/project/resource_gallery_of_interventions/page/StressThermometer`. [Accessed 2022].

[16] Gorski, T.: The Stress Thermometer. Available online: `https://i.redd.it/jutk7qv4nkb41.jpg`.

[17] Stress Level Thermometer. Available online: `https://www.etsy.com/listing/1110274677/stress-level-thermometer-pdf`. [Accessed 2022].

[18] SUDS Thermometer. Centre for Clinical Psychology. Available online: `https://ccp.net.au/suds-thermometer/`.

[19] Wolpe, J.: The Practice of Behavior Therapy. Pergamon Press, 1969, pp. 314.

[20] Koo, T. K.—Li, M. Y.: A Guideline of Selecting and Reporting Intraclass Correlation Coefficients for Reliability Research. Journal of Chiropractic Medicine, Vol. 15, 2016, No. 2, pp. 155–163, doi: 10.1016/j.jcm.2016.02.012.

[21] Gamer, M.—Lemon, J.—Fellows, I.—Singh, P.: irr: Various Coefficients of Interrater Reliability and Agreement. R package version 0.84.1. Available online: `https://CRAN.R-project.org/package=irr`. [Accessed 2022].

[22] Boersma, P.—Weenink, D.: Praat: Doing Phonetics by Computer [Computer

program; Version 6.2.06]. 1992–2022. Available online: https://www.praat.org. [Accessed January 2022].

[23] Traunmüller, H.—Eriksson, A.: Acoustic Effects of Variation in Vocal Effort by Men, Women, and Children. The Journal of the Acoustical Society of America, Vol. 107, 2000, No. 6, pp. 3438–3451, doi: 10.1121/1.429414.

[24] Švec, J. G.—Granqvist, S.: Tutorial and Guidelines on Measurement of Sound Pressure Level in Voice and Speech. Journal of Speech, Language, and Hearing Research, Vol. 61, 2018, No. 3, pp. 441–461, doi: 10.1044/2017_JSLHR-S-17-0095.

[25] Liénard, J. S.: Quantifying Vocal Effort from the Shape of the One-Third Octave Long-Term-Average Spectrum of Speech. The Journal of the Acoustical Society of America, Vol. 146, 2019, No. 4, pp. EL369-EL375, doi: 10.1121/1.5129677.

[26] Kirchhübel, C.—Howard, D. M.—Stedmon, A. W.: Acoustic Correlates of Speech when Under Stress: Research, Methods and Future Directions. International Journal of Speech Language and the Law, Vol. 18, 2011, No. 1, pp. 75–98, doi: 10.1558/ijsll.v18i1.75.

[27] Ruiz, R.—Absil, E.—Harmegnies, B.—Legros C.—Poch, D.: Time- and Spectrum-Related Variabilities in Stressed Speech Under Laboratory and Real Conditions. Speech Communication, Vol. 20, 1996, No. 1-2, pp. 111–129, doi: 10.1016/S0167-6393(96)00048-9.

[28] Van Puyvelde, M.—Neyt, X.—McGlone, F.—Pattyn, N.: Voice Stress Analysis: A New Framework for Voice and Effort in Human Performance. Frontiers in Psychology, Vol. 9, 2018, Art. No. 1994, doi: 10.3389/fpsyg.2018.01994.

[29] Kejriwal, J.—Beňuš, Š.—Trnka, M.: Stress Detection Using Non-Semantic Speech Representation. Proceedings of the 2022 32nd International Conference Radioelektronika, Košice, Slovakia, 2022, pp. 1–5, doi: 10.1109/RADIOELEKTRONIKA54537.2022.9764916.

[30] Pedregosa, F.—Varoquaux, G.—Gramfort, A.—Michel, V.—Thirion, B.—Grisel, O. et al.: Scikit-Learn: Machine Learning in Python. Journal of Machine Learning Research, Vol. 12, 2011, pp. 2825–2830, https://jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf.

[31] Eyben, F.—Wöllmer, M.—Schuller, B.: Opensmile: The Munich Versatile and Fast Open-Source Audio Feature Extractor. Proceedings of the 18th ACM International Conference on Multimedia (MM '10), 2010, pp. 1459–1462, doi: 10.1145/1873951.1874246.

[32] Eyben, F.—Scherer, K. R.—Schuller, B. W.—Sundberg, J.—André, E.—Busso, C.—Devillers, Y. L.—Epps, J.—Laukka, P.—Narayanan, S. S.—Truong, K. P.: The Geneva Minimalistic Acoustic Parameter Set (GeMAPS) for Voice Research and Affective Computing. IEEE Transactions on Affective Computing, Vol. 7, 2016, No. 2, pp. 190–202, doi: 10.1109/TAFFC.2015.2457417.

[33] Snyder, D.—Garcia-Romero, D.—Sell, G.—Povey, D.—Khudanpur, S.: X-Vectors: Robust DNN Embeddings for Speaker Recognition. Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 5329–5333, doi: 10.1109/ICASSP.2018.8461375.

[34] Shor, J.—Jansen, A.—Maor, R.—Lang, O.—Tuval, O.—de Chaumont

Quitry, F.—Tagliasacchi, M.—Shavitt, I.—Emanuel, D.—Haviv, Y.: Towards Learning a Universal Non-Semantic Representation of Speech. Proceedings of INTERSPEECH 2020, 2020, pp. 140–144, doi: 10.21437/Interspeech.2020-1242.

[35] Povey, D.—Ghoshal, A.—Boulianne, G.—Burget, L.—Glembek, O.—Goel, N.—Hannemann, M.—Motlícek, P.—Qian, Y.—Schwarz, P.—Silovský, J.—Stemmer, G.—Veselý, K.: The Kaldi Speech Recognition Toolkit. Proceedings of the IEEE 2011 Workshop on Automatic Speech Recognition and Understanding, 2011.

[36] Nagrani, A.—Chung, J. S.—Zisserman, A.: VoxCeleb: A Large-Scale Speaker Identification Dataset. Proceedings of INTERSPEECH 2017, 2017, pp. 2616–2620, doi: 10.21437/Interspeech.2017-950.

[37] Chung, J. S.—Nagrani, A.—Zisserman, A.: VoxCeleb2: Deep Speaker Recognition. Proceedings of INTERSPEECH 2018, 2018, pp. 1086–1090, doi: 10.21437/Interspeech.2018-1929.

[38] Shor, J.—Jansen, A.—Maor, R.—Lang, O.—Tuval, O.—de Chaumont Quitry, F.—Tagliasacchi, M.—Shavitt, I.—Emanuel, D.—Haviv, Y.: Nonsemantic-Speech-Benchmark/Trill. Available online: `https://aihub.cloud.google.com/p/products%2F41239b97-c960-479a-be50-ae7a23ae1561`. [Accessed 2022].

[39] Barros, P.—Churamani, N.—Lakomkin, E.—Siqueira, H.—Sutherland A.—Wermter, S.: The OMG-Emotion Behavior Dataset. CoRR, 2018, doi: 10.48550/arXiv.1803.05434.

[40] Rusko, M.—Darjaa, S.—Trnka, M.—Ritomský, M.—Sabo, R.: Alert! ... Calm Down, There Is Nothing to Worry About. Warning and Soothing Speech Synthesis. Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014), Reykjavík, 2014, pp. 1182–1187, `http://www.lrec-conf.org/proceedings/lrec2014/pdf/722_Paper.pdf`.

[41] Guoth, I.—Rusko, M.—Ritomský, M.—Trnka, M.—Darjaa, S.: Identifying Tense Arousal in Speech Using Phase Based Features. Proceedings of Meetings on Acoustics, Vol. 30, 2017, No. 1, Art. No. 06005, doi: 10.1121/2.0000659.

Marián Trnka is a researcher and coder at the Institute of Informatics, Slovak Academy of Sciences. His expertise lies in programming, artificial intelligence, machine learning, natural language processing, speech synthesis, speech, speaker, and emotion recognition. He earned his Master's degree in cybernetics from the Slovak Technical University in Bratislava in 1994. He has authored more than 90 scientific papers and has contributed to applications such as the Automatic Dictation System for Judicial Domain (APD), automatic broadcast news subtitling and interactive fairy-tale reading application Readmio.

Sakhia Darjaa is a researcher and coder at the Institute of Informatics, Slovak Academy of Sciences. He specializes in programming, artificial intelligence, machine learning, natural language processing, speech synthesis, speech signal analysis, speaker, and emotion recognition. He holds his Master's degree in automatic systems of control from the Technical University in Košice, awarded in 1980, and obtained his Doctoral degree in telecommunications from the same institution in 2013. He is the author of more than 90 scientific papers and has been involved in several projects aimed at designing applications.

Róbert Sabo is a researcher and linguist affiliated with the Institute of Informatics, Slovak Academy of Sciences. His expertise encompasses linguistic analysis, phonetics, prosody research, speech corpus development, and expressive speech analysis. He earned his Master's degree in Slovak language and literature from the Comenius University in Bratislava in 2007. He completed his Doctoral degree in linguistics at the Ľudovít Štúr Institute of Linguistics of the Slovak Academy of Sciences in 2013. He is the author and co-author of over 40 scientific papers in the field of speech processing and computational linguistics.

Milan Rusko serves as the Head of the Department of Speech Analysis and Synthesis and is a project manager and researcher at the Institute of Informatics, Slovak Academy of Sciences. His expertise is in acoustics, artificial intelligence, speech synthesis, speech analysis, speaker recognition, and emotion recognition. He obtained his Master's degree in radio-electronics from the Technical University in Bratislava in 1984 and later earned his Doctoral degree in telecommunications from the Technical University in Košice in 2013. He has authored more than 120 scientific papers in the field of speech processing and has held positions on the Board of the Slovak Acoustic Society.

Meilin Schaper joined DLR's Institute of Flight Guidance in 1997 as an applications programmer and received her diploma in computer science from the University of Hagen in 2003. Since 2003 she worked on the concept and tool development of the DLR/Eurocontrol DMAN research prototype supporting tower air traffic controllers in departure sequence optimization. She was a project manager of the continued work, enhancing the tool (now called CADEO), and integrating and validating it in shadow mode trials. Further research topics include airport collaborative decision making, total airport management and combining planning systems to support operators and controllers seamlessly. Currently, her main research interest is ATM security.

Tim Stelkens-Kobsch received his diploma in aeronautical engineering from the University of Braunschweig in 2001, where he continued scientific activities as a researcher and teaching assistant for Air Traffic Management. He has extensive experience in the fields of guidance and control, controller assistance systems, and simulation and validation. In 2010 he joined the DLR's Institute of Flight Guidance in Braunschweig. Within the DLR he works on ATM-Simulation and Aviation Security and is responsible for the Generic Cockpit Simulator (GECO). He headed several validations regarding Air Traffic Management and has extensive experience in the management of international projects.

Computing and Informatics, Vol. 42, 2023, 1255–1280, doi: 10.31577/cai_2023_5_1255

MULTI-LABEL BIRD SPECIES CLASSIFICATION USING SEQUENTIAL AGGREGATION STRATEGY FROM AUDIO RECORDINGS

Noumida Abdul Kareem

College of Engineering Trivandrum
APJ Abdul kalam Technological University, Thiruvananthapuram, India
e-mail: noumidaa@gmail.com

Rajeev Rajan

Government Engineering College, Barton Hill,Thiruvananthapuram
APJ Abdul kalam Technological University, Thiruvananthapuram, India
e-mail: rajeev@cet.ac.in

Abstract. Birds are excellent bioindicators, playing a vital role in maintaining the delicate balance of ecosystems. Identifying species from bird vocalization is arduous but has high research gain. The paper focuses on the detection of multiple bird vocalizations from recordings. The proposed work uses a deep convolutional neural network (DCNN) and a recurrent neural network (RNN) architecture to learn the bird's vocalization from mel-spectrogram and mel-frequency cepstral coefficient (MFCC), respectively. We adopted a sequential aggregation strategy to make a decision on an audio file. We normalized the aggregated sigmoid probabilities and considered the nodes with the highest scores to be the target species. We evaluated the proposed methods on the Xeno-canto bird sound database, which comprises ten species. We compared the performance of our approach to that of transfer learning and Vanilla-DNN methods. Notably, the proposed DCNN and VGG-16 models achieved average F1 metrics of 0.75 and 0.65, respectively, outperforming the acoustic cue-based Vanilla-DNN approach.

Keywords: Multi-label, sequential, augmentation, recurrent neural network, convolutional neural network, transfer learning

1 INTRODUCTION

Over the last few decenniums, significant research efforts have been devoted to automatic speech analysis. However, there has recently been an upsurge in the study towards the automated analysis of animal and avian vocalizations. Bird detection is critical for avian biodiversity conservation because it allows ornithologists to count the number of birds in a particular location. A bird may listen to other birds and determine whether they are neighbours or strangers, kin or non-kin.

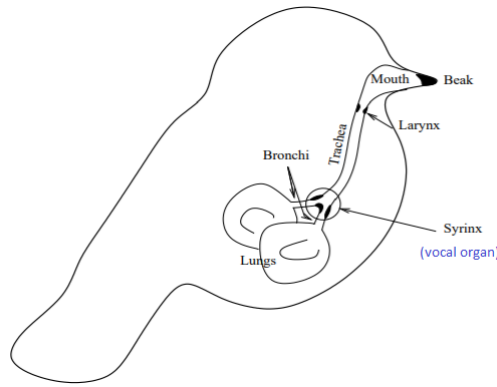


Figure 1. Avian sound production [1]

Figure 1 illustrates the avian sound production system [1]. In birds, the sound production mechanism consists of the lungs, bronchi, syrinx, trachea, larynx, mouth, and beak [1]. The vocal tract modulates airflow from the lungs as it traverses through the bronchi to the syrinx. The trachea is made of complete cartilage rings. Complete C-shaped cartilage rings with open ends against each other make up the bronchial elements. Airflow causes the syringeal medial tympaniform membrane (MTM) in each bronchus to vibrate nonlinearly opposite the cartilage wall during a bird's song. In birds, the mouth functions as a cavity resonator in the same way that it does in humans, but it is less flexible. Both bird calls and songs are examples of bird vocalization.

Bird songs are more intricate vocalizations than bird calls, which are thought to be simple[1]. Typically, songs are produced spontaneously by the male. The hierarchical levels of a bird's song consist of phrases, syllables, and elements. When a bird changes the order of the phrases in the songs, it can generate diverse singing types. Bird calls are short and are produced by connecting a series of sounds [1]. Figure 2 depicts the vocalization patterns of Eurasian Owl and Red-wattled Lapwing through mel-spectrograms.

[1] https://en.wikipedia.org/wiki/Bird-vocalization

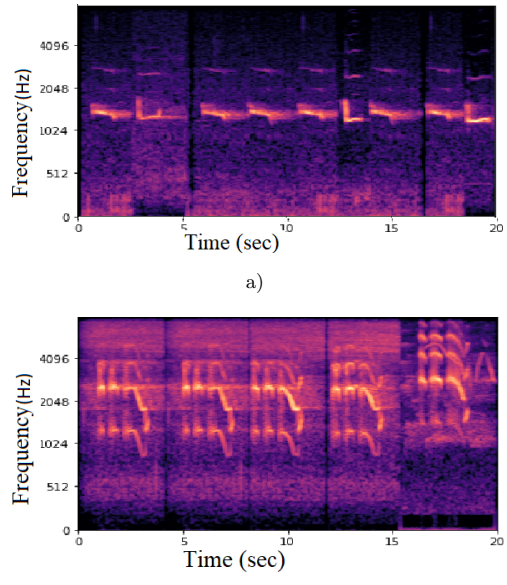


Figure 2. Mel-spectrograms of vocalizations of Eurasian Owl a), Red. Lapwing b)

Traditional field methods for tracking and identifying various bird species require significant human labour. The Global Biodiversity Information Facility (GBIF)[2], which builds biological multimedia databases, also works on automatic species identification from field recordings. Acoustic bird monitoring is an effective strategy since most birds communicate primarily through vocalizations [2]. Some of the speech and audio processing techniques for the recognition of bird calls can be referred to in [3, 4, 5].

Despite weather noise and a wide variety of bird call types, machine learning approaches, particularly deep learning, can obtain very high recognition rates on remote monitored auditory data [6]. There have been numerous endeavours in the literature to classify birds, from pre-segmented acoustic single-label audio recordings [7, 8, 9, 10, 11]. Multi-label bird classification is difficult because of the time-frequency overlapping in the audio recordings. A bag generator is proposed to convert an audio recording into a bag-of-instances representation, followed by a multi-instance multi-label (MIML) classifier to forecast the set of species present in the recording [12]. It is formulated as a problem in the MIML framework for supervised classification. A multi-label classification model for finding simultaneous auditory patterns in long-duration recordings is proposed in [13]. Some of the previous works in multi-label bird call classification include [14, 15, 16, 17].

[2] `https://www.gbif.org`

The algorithm in [18] employs a deep learning technique based on convolution layers to predict the most dominant foreground species in an acoustic scenario. While forecasting the major species of each sound file, the network design yields a mean average precision (MAP) score of 0.686. The efficacy of various CNN-derived features for detecting bird vocalization is explored in [19]. Due to the difficulties in acquiring annotated training sounds, the utilization of transfer learning in CNN might be advantageous in bird call classification. An efficient CNN-based transfer learning approach for bird-call identification is explored in [20]. A particularly challenging task of bio-acoustic classification pertains to detecting overlapping events in an acoustic scene. In this context, our work aims to discern both multiple simultaneous and isolated bird vocalizations in audio recordings.

The main contributions of the paper are:

1. The sequential aggregation strategy has been implemented effectively on MFCC and mel-spectrogram features for bird call identification.
2. We present a DCNN model for mel-spectrogram inputs and systematically compare its performance with RNN, Vanilla-DNN, and transfer learning schemes.
3. SpecAugment-based data augmentation scheme (time masking, frequency masking and time warping) has been implemented for creating additional training files for the network.

A brief overview of the multi-label classification system is provided in Section 2. The performance evaluation, including the detailed dataset description, is explained in Section 3, followed by the result analysis in Section 4. Finally, the paper is concluded in Section 5.

2 SYSTEM DESCRIPTION

We proposed two sequential aggregation models for multi-label bird species classification, namely, the Acoustic RNN/DNN models and the mel-spectrogram CNN models. Features namely, mel-frequency cepstral coefficients (MFCC) and mel-spectrograms are extracted. MFCC features are queried to Acoustic RNN/DNN models (LSTM, GRU and Vanilla-DNN), whereas mel-spectrogram features to mel-spectrogram CNN models (pre-trained models, proposed DCNN) for classification. Multi-label classification is performed using a novel sequential aggregation strategy. The scheme for detecting the vocalization is illustrated in Figure 3.

2.1 Acoustic RNN Models

MFCCs find extensive application in diverse audio classification tasks driven by human perception [21], serving as predictors of timbre similarity perception [22]. MFCC converts the raw audio data into a compact and informative representation that captures the relevant information in the signal while removing irrelevant or redundant information. Gated Recurrent Unit (GRU) and Long Short-Term Memory

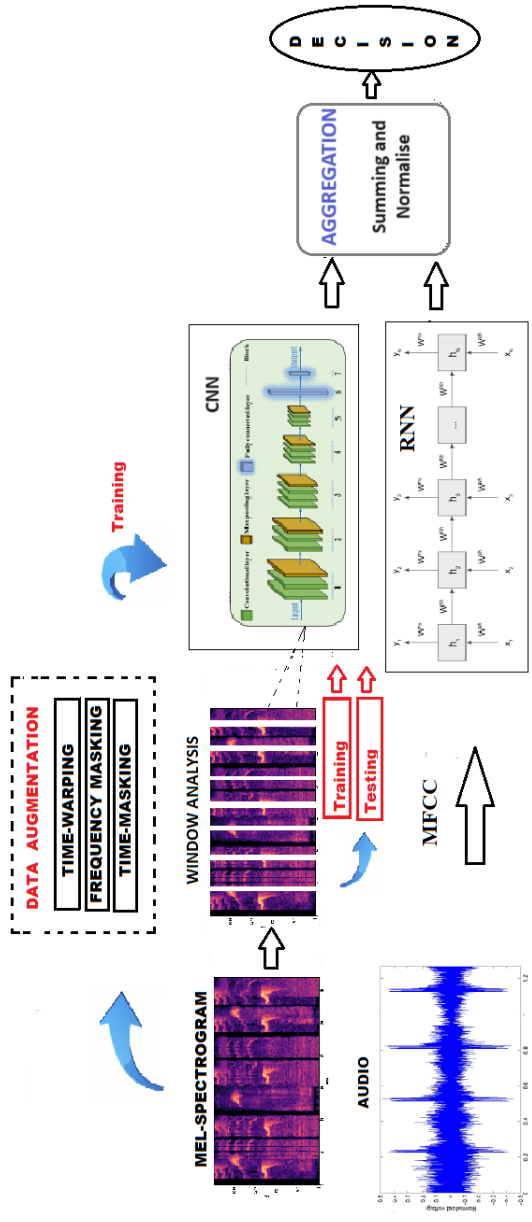


Figure 3. Block diagram: proposed multi-label bird classification using sequential aggregation strategy

(LSTM) stand out as two popular variants of Recurrent Neural Networks (RNN) that possess long-term memory capabilities. By effectively storing past inputs within their internal states and leveraging this historical context to target vectors, these networks excel at processing sequential data and retaining valuable memories. The present study delves into the examination of LSTM and GRU, exploring their capacity to capture long-range dependencies and learn temporal patterns using MFCCs. These models were trained using our multi-label dataset and implemented using sequential aggregation strategy.

2.1.1 Sequential LSTM

The LSTM cell, a recurrent network node equipped with an input, output, and a forget gate to mitigate vanishing gradients [23, 24], is harnessed to efficiently capture temporal patterns inherent in audio, as illustrated in Table 1. By leveraging frame-wise computed MFCC, LSTM effectively taps into the sequential nature of the data. LSTM can memorise previous data and predict the future with the aid of the information stored in the memory [25].

No.	**Output Shape**	**Description**
1	(None, 64, 1 048)	LSTM, 1 048 hidden nodes
2	(None, 64, 728)	LSTM, 728 hidden nodes
3	(None, 432)	LSTM, 432 hidden nodes
4	(None, 432)	Dropout, 0.2
5	(None, 10)	Dense, 10 hidden nodes

Table 1. LSTM architecture

An LSTM can be mathematically represented as follows:

$$u_i = \tanh(\xi_{xu} * x_i + \xi_{hu} * h_{i-1} + d_u), \tag{1}$$

$$r_i = \sigma(\xi_{xr} * x_i + \xi_{hr} h_{i-1} + d_r), \tag{2}$$

$$f_i = \sigma(\xi_{xf} * x_i + \xi_{hf} * h_{i-1} + d_f), \tag{3}$$

$$o_i = \sigma(\xi_{xo} * x_i + \xi_{ho} * h_{i-1} + d_o), \tag{4}$$

$$c_i = r_i u_i + f_i c_{i-1}, \tag{5}$$

$$h_i = \tanh(c_i o_i), \tag{6}$$

$$output_{class} = \sigma(h_i * \xi_{outpara}), \tag{7}$$

where u_i, r_i, f_i, o_i, c_i represents update equations for input gate, forget gate, output gate, cell state and cell output, respectively. ξ_{xu}, ξ_{xr}, ξ_{xf}, ξ_{xo} and ξ_{hu}, ξ_{hr}, ξ_{hf}, ξ_{ho}, $\xi_{outpara}$ are weights, and d_u, d_r, d_f, d_o are biases to be computed during training. The output of a neuron at time i is denoted as h_i, and elementwise multiplication is represented by '.'. The activation functions employed in our model are $\sigma(.)$ for the

sigmoid function and $tanh(.)$ for the hyperbolic tangent function. x_i denotes the input feature vector at time i. $output_{class}$ is the classification output.

2.1.2 Sequential GRU

In the realm of sequence modelling techniques, GRU emerges as the latest addition, succeeding RNN and LSTM, and thereby holding the promise of enhancing various sequential processing applications. RNNs have gained widespread adoption in language recognition [26] due to their ability to process sequential data effectively. For deep sentence processing, different cell types have been devised to improve neural networks' ability to capture long-term dependencies. The fundamental difference between GRU and LSTM neural network architectures resides in their gate structures. GRU, characterized by its simplicity, features two primary gates: the reset gate and the update gate. In contrast, LSTM, a more complex architecture, incorporates three distinct gates into its design. The GRU cells exhibit comparable power to LSTM cells [27], especially for smaller data sets, while requiring fewer computational resources.

No.	**Output Shape**	**Description**
1	(None, 64, 1 048)	GRU, 1 048 hidden nodes
2	(None, 768)	GRU, 728 hidden nodes
3	(None, 10)	Dense, 10 hidden nodes

Table 2. GRU architecture

The governing equations for GRU is presented as follows [28]:

$$z_i = \sigma_g(W_z.x_i + U_z.h_{i-1} + b_z), \tag{8}$$

$$r_i = \sigma_g(W_r.x_i + U_r.h_{i-1} + b_r), \tag{9}$$

$$\hat{h_i} = \phi_h(W_h.x_i + U_h.(r_i \odot h_{i-1}) + b_h), \tag{10}$$

$$h_i = (1 - z_i) \odot h_{i-1} + z_i \odot \hat{h_i}. \tag{11}$$

Here, the variables x_i, h_i, $\hat{h_i}$, g_i, and f_i represent the input, output, candidate activation, update gate, and reset gate vectors, respectively. The matrices W, U, and b represent parameter matrices, while σ_g and ϕ_h denote the activation functions. The symbol $\odot$ denotes the Hadamard product, and '.' signifies elementwise multiplication. The GRU architecture utilized in the proposed work is conveniently summarized in Table 2.

2.2 Mel-Spectrogram CNN Models

A mel-spectrogram is a visual representation that shows how the frequencies of a signal change over time [29, 30]. It uses a special filter called a mel-scale filter bank

to highlight the frequencies most important for human perception. The mel scale was developed to scale frequency data in a manner that more closely resembles how humans perceive sound. Mels are units on the mel scale, and a reference frequency of 1 000 Hz, 40 dB above a listener's threshold is defined as 1 000 mels. The number of mels associated with a tone closely corresponds to its frequency below 500 Hz. Above 500 Hz, the number of mels between pitches perceived as "evenly spaced" increases as frequency increases. The mel-spectrogram is a smoothed spectrogram with highly emphasized low-frequency components. Here mel-spectrogram is computed with 128 bins and a frame size of 40 ms and a hop size of 10 ms. Figure 4 depicts mel-spectrograms of two audio files containing two and three species.

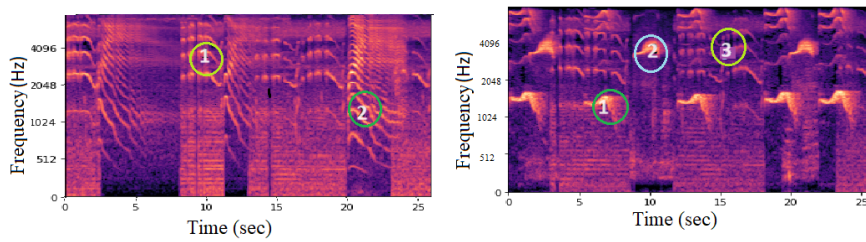


Figure 4. Mel-spectrogram of bird's vocalization with multiple bird sounds in a single audio recording. Repetitive patterns in the vocalization are shown in circles for 2 species a) and 3 species b).

CNN has been extensively used as one of the representation learning methods that enable a machine to automatically detect the representations or patterns required for classification tasks [31]. We utilized two CNN architectures for the sequential aggregation model. One is based on transfer learning models and the other one is based on a proposed DCNN shown in Table 3. Data augmentation techniques such as time-warping, frequency masking, and time-masking are used to create additional mel-spectrograms during the training phase of the network

2.2.1 Sequential DCNN

The architecture shown in Table 3 is used in the proposed analysis. The resulting $432 \times 1\,008$ arrays of the mel-spectrograms after data preparation were fed into the CNN model. The model comprises multiple convolutional layers with 3×3 kernels, 'same' padding, and 'glorot_uniform' kernel initialization, followed by Leaky ReLU activation (LeakyReLU (alpha = 0.33)). Maxpooling layers with 3×3 pool size and stride 3 are inserted after each pair of convolutional layers. Dropout layers with a 0.25 dropout rate follow maxpooling layers. After every two layers, the number of channels for the convolution layer is increased by a multiple of two, from 32 to 256. The model concludes with a GlobalMaxPooling2D layer to reduce spatial

dimensions, two dense layers (Dense(1 024) and Dense(10, activation = 'sigmoid')) for classification, and Leaky ReLU activation in the first dense layer.

No.	**Input Shape**	**Description**
1	$3 \times 432 \times 1\,008$	Mel-spectrogram
2	$32 \times 434 \times 1\,010$	$32 \times 3 \times 3$ Conv1
3	$32 \times 436 \times 1\,012$	$32 \times 3 \times 3$ Conv2
4	$32 \times 145 \times 337$	3×3 Maxpooling, Dropout (0.25)
5	$64 \times 147 \times 339$	$64 \times 3 \times 3$ Conv3
6	$64 \times 149 \times 341$	$64 \times 3 \times 3$ Conv4
7	$64 \times 49 \times 113$	3×3 Maxpooling, Dropout (0.25)
8	$128 \times 51 \times 115$	$128 \times 3 \times 3$ Conv5
9	$128 \times 53 \times 117$	$128 \times 3 \times 3$ Conv6
10	$128 \times 17 \times 39$	3×3 Maxpooling, Dropout (0.25)
11	$256 \times 19 \times 41$	$256 \times 3 \times 3$ Conv7
12	$256 \times 21 \times 43$	$256 \times 3 \times 3$ Conv8
13	$256 \times 1 \times 1$	GlobalMaxPooling2D
14	1 024	Fully connected (Flatten)
15	1 024	Dropout (0.5)
16	10	Sigmoid

Table 3. Proposed DCNN architecture

The equations for the Leaky ReLU activation (LReLU) and sigmoid activation are:

Leaky ReLU:

$$\mathrm{LReLU}(a) = \begin{cases} a, & \text{if } a > 0, \\ 0.33a, & \text{if } a \leq 0, \end{cases}$$

$$\mathrm{Sigmoid}(a) = \frac{1}{1 + \exp(-a)}.$$

These activations introduce non-linearity in the model. The Leaky ReLU helps mitigate the vanishing gradient problem, while the sigmoid activation in the final layer maps the model's output to a range between 0 and 1 for multi-class classification.

To learn a non-linear function from input to output that generalises well and yields good classification accuracy on unknown data, CNN heavily relies on the availability of massive amounts of training data [32]. Data augmentation, which involves deforming a set of annotated training files to produce additional training data, is an elegant solution to this problem. We adopted SpecAugment [33] as a data augmentation technique for the proposed scheme.

2.2.2 Sequential Transfer Learning

In transfer learning, a model created for one application is customized for another task. It is a popular deep learning approach that commences with pre-trained models for pattern recognition and computer vision tasks. We experimented with five pre-trained networks for the proposed task, namely VGG-16, ResNet50, InceptionV3, InceptionResNetV2, and Efficient-NetB3. These models were re-trained using our multi-label dataset and implemented using sequential aggregation strategy. The details of the baseline pre-trained models are presented in Table 4.

No.	**Model**	**Parameters**	**Layers**	**Activation**
1	**InceptionResNetV2**	56 M	164	ReLU
2	**InceptionV3**	23.9 M	48	ReLU
3	**VGG-16**	138 M	16	ReLU
4	**Efficient-NetB3**	12 M	300	ReLU
5	**ResNet50**	25.6 M	50	ReLU

Table 4. Model description (M-Million)

VGG-16. The deep convolutional VGG-16 model is retrained in this experiment to detect multiple species. Table 5 shows the VGG-16 architecture, including 13 convolutions and three fully connected layers. The convolution layers are all 3×3 layers, with the same padding and stride size of 1, and the pooling layers are all 2×2 layers, with a stride size of 2. After data preparation, the resulting $432 \times 1\,008$ arrays of the mel-spectrograms are resized to 256×256 pixels. Before the fully connected layers, the last feature map has 512 channels and is flattened into a vector with 32 768 values. Finally, the dense layer with 4 096 neurons is used to add the fully connected layers, followed by a dropout layer with a value of 0.5. The proposed VGG-16 architecture for our experiment requires ten classes. The sigmoid function is chosen in the output layer.

ResNet50. Residual Networks (ResNet) are a type of deep neural network commonly used as the framework for many computer vision applications. ResNet50 is a 50-layer DCNN architecture with 48 convolutional layers, one maxpooling layer, and one average pooling layer. It is a variant of the ResNet model that uses residual learning [20]. To solve the vanishing gradient problem, the concept called residual network architecture is introduced. ResNet50 uses skip connections to propagate the activations to reduce the impact.

InceptionV3. The InceptionV3 is a 48-layer deep CNN architecture. Convolutions, poolings, dropouts, and fully connected layers make up the model. Sigmoid loss is computed and batch normalization is used throughout the model [34].

InceptionResNetV2. InceptionResNetV2, a 164-layer deep CNN architecture based on the Inception family's design but with residual linkages, is a variation of InceptionV3. The number of auxiliary classifiers is reduced from three to two.

No.	**Input Shape**	**Description**
1	$3 \times 256 \times 256$	Conv 3×3 ($\times 2$), Stride $= 1$
2	$64 \times 256 \times 256$	Maxpooling 2×2, Stride $= 2$
3	$64 \times 128 \times 128$	Conv 3×3 ($\times 2$), Stride $= 1$
4	$128 \times 128 \times 128$	Maxpooling 2×2, Stride $= 2$
5	$128 \times 64 \times 64$	Conv 3×3 ($\times 3$), Stride $= 1$
6	$256 \times 64 \times 64$	Maxpooling 2×2, Stride $= 2$
7	$256 \times 32 \times 32$	Conv 3×3 ($\times 3$), Stride $= 1$
8	$512 \times 32 \times 32$	Maxpooling 2×2, Stride $= 2$
9	$512 \times 16 \times 16$	Conv 3×3 ($\times 3$), Stride $= 1$
10	$512 \times 16 \times 16$	Maxpooling 2×2, Stride $= 2$
11	32 768	Fully connected (Flatten)
12	4 096	Dropout (0.5)
13	4 096	Dropout (0.5)
14	10	Sigmoid

Table 5. VGG-16 architecture

EfficientNet. EfficientNet is a CNN model that uses a compound coefficient to scale all width/depth/resolution dimensions uniformly. There are eight models in the EfficientNet architecture group, ranging from B0 to B7. Each model number denotes a variant with greater precision and a greater number of parameters [35]. To significantly reduce calculation costs while maintaining accuracy, EfficientNet divides the original convolution into two stages: depthwise and pointwise convolution. Because of its linear bottleneck architecture, which uses linear activation in the final layer of each block to prevent data loss from ReLU, the network is efficient.

2.3 Vanilla-DNN Model

A Vanilla-DNN framework is also used for performance comparison. The Librosa Python library is used to compute 40-dimensional MFCCs in the front-end.

Our model has two hidden layers, each with 256 perceptrons, followed by the ReLU activation function and a 50 % dropout. The sigmoid activation function is chosen in the output layer. The model is trained using our multi-label dataset and implemented using sequential aggregation strategy.

3 PERFORMANCE EVALUATION

3.1 Data Set

Recordings of the bird species are collected from the Xeno-canto website [36][3]. We standardized all the files to a minimum sampling rate of 16 kHz because the original

[3] `https://www.xeno-canto.org` (Xeno-canto)

files' sample rates ranged from 16 kHz to 44.1 kHz.

Details of scientific names and the number of Xeno-canto files used for each bird species are illustrated in Table 6. Table 7 gives the dataset specification after pre-processing. The train set contains 1 078 isolated audio files of 10 species. The files are refined such that one vocalization of 1.5 s duration is in each audio file. HC (House Crow-111), MD (Mallard Duck-106), AK (Asian Koel-121), EO (Eurasian Owl-107), HS (House Sparrow-100), BJ (Blue Jay-109), RL (Red-wattled Lapwing-104), GG (Grey Go-away-109), IP (Indian Peafowl-103), and WW (Western Wood-Pewee-108) are among the birds featured. The names and number of audio files after pre-processing are indicated in brackets. The test set contains 434 audio files that contain overlapping and multiple calls, often consisting of 2 or 3 distinct vocalizations.

No.	**Code – Scientific Name**	**# XC**	**Specialities**
1	HC – Corvus splendens	27	loud, cawing call, "caw"
2	MD – Anas platyrhynchos	25	low pitch "quacks", "whistles", "grunts", "hank"
3	AK – Eudynamys chinensis	26	loud, repetitive cuckoo-like call, "cooing"
4	EO – Bubo bubo	25	deep haunting, hooting call, "hoo-hoo-hoo"
5	HS – Passer domesticus	24	cheerful, trilling, chirping call, "chip"
6	BJ – Cyanocitta cristata	27	"jay jay" or "scold-call", "chak", "wheeoo"
7	RL – Vanellus vanellus	24	loud, wailing call
8	GG – Corythaixoides concolor	19	loud, honking, clear-territorial call, "kwaa"
9	IP – Pavo cristatus	29	piercing screams, "gobbling", "peacock"
10	WW – Contopus sordidulus	24	loud, clear whistle, peenting "pee-a-wee"

Table 6. Details of Xeno-Canto (XC) files

No.	**Class**	**Count (Bird Files)**	**# Calls**
1	Audio Files (Train)	1 078	1 078
2	Audio Files (Test)		
	Calls with two species	334	668
	Calls with three species	100	300
	Total	1 512	**2 046**

Table 7. Dataset specification

The selection is based on some below-mentioned rules. Firstly, the ten selected species represent a broad range of bird call patterns well-defined in previous

works [37, 38]. The bird call structures mainly consist of chirp, whistle, block, warbles, and click. This renders the proposed system to satisfy the generic requirement. Secondly, the selected species should have adequate samples to train and test the proposed method.

3.2 Data Augmentation

As cited in [39, 40], CNN's efficacy is highly reliant on abundant data for achieving superior results, which can be limited when dealing with small data sizes. To address this limitation and enhance the training process, data augmentation techniques are employed. In the proposed scheme, SpecAugment [33] serves as the augmentation strategy, involving the masking of frequency channels and time frames within the mel-spectrogram image representation. This augmentation includes time warping, frequency masking, and time masking of mel-spectrograms, as illustrated in Figure 5.

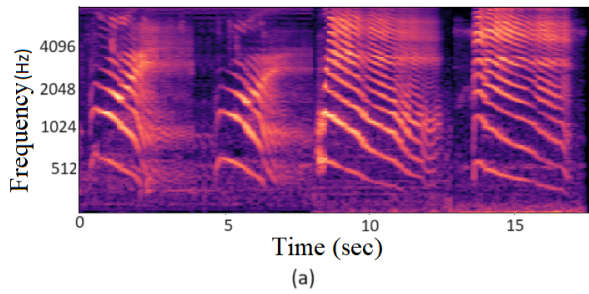
A log mel-spectrogram, comprising τ time steps, can be visualized as an image, with the time axis running horizontally and the frequency axis running vertically. Within the time steps ranging from W to $(\tau - W)$, a point randomly selected along the central horizontal line of the mel-spectrogram can undergo a warp to the left or right, covering a distance of W. Additionally, this visualization incorporates frequency masking and time masking techniques, where certain frequency bands and time segments are selectively masked. To enrich the train set, we generated 8 923 mel-spectrograms for the DCNN model and an additional 3 344 mel-spectrograms as augmented data for the transfer learning models [33].

3.3 Sequential Aggregation Strategy

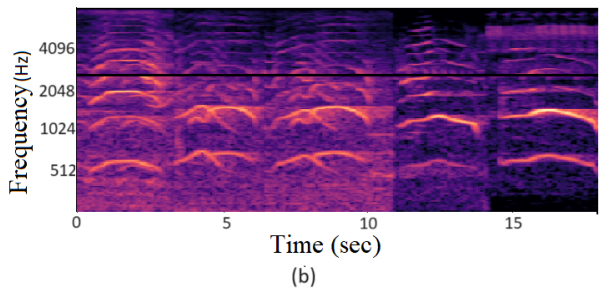
Audio recordings are sliced into fixed-length segments. For acoustic models, MFCCs of sliced audios are extracted and fed to RNN/DNN models. Sliced mel-spectrograms are utilized for training and testing with DCNN and pre-trained models. The model trained on ten classes is used to predict the probability of ten bird species. The trained neural network then predicts the probability of each species present in a segment. We used an aggregation strategy to decide on the test data. Since multiple species exist per audio clip, multiple sigmoid outputs from slices are aggregated and normalized. The nodes corresponding to the highest probability values are considered the target species. Figure 6 illustrates the schematic of the sequential aggregation process used in all the proposed methods.

3.4 Experimental Framework

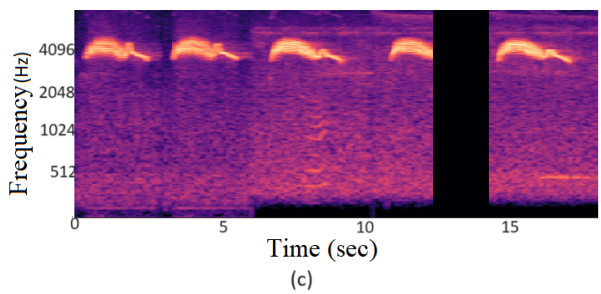
MFCC and mel-spectrogram features are extracted using Librosa Python package. Proposed DCNN model, pre-trained models (VGG-16, ResNet50, InceptionV3, InceptionResNetV2, and EfficientNetB3), DNN and RNN-based models (GRU and



a) Time warped



b) Frequency masked



c) Time masked

Figure 5. SpecAugment-generated mel-spectrograms of bird calls

LSTM) are implemented using the novel sequential aggregation strategy. Additionally, some existing models in the literature are also implemented using sequential aggregation strategy. All these models were trained on a Google Colab notebook.

Audio files are converted to a time-frequency representation using short-time Fourier transform (STFT) with 480 samples for the window. The mel-spectrogram is segmented into small duration chunks and fed to DCNN. Adaptive moment estimation (Adam) was used in the training process of the network for optimizing the

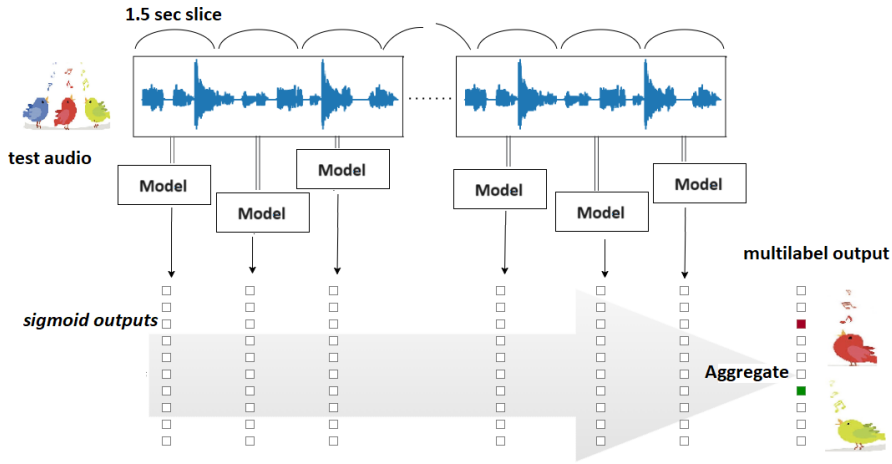


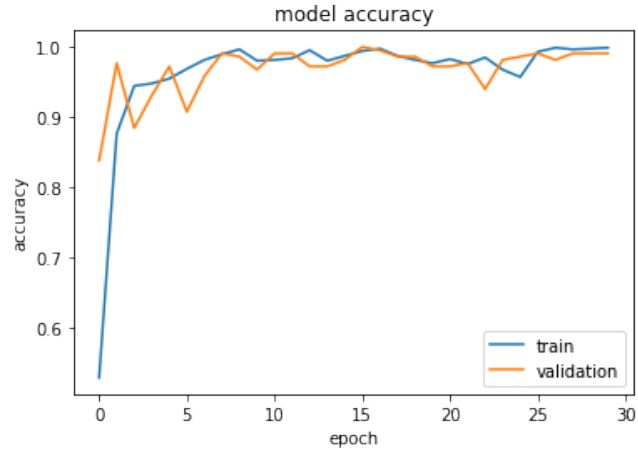
Figure 6. Schematic of window sliding technique used in the experiments

categorical cross-entropy between predictions and targets. After each maxpooling layer, the training was regularized with dropouts at 0.25. The sigmoid activation function was used and the model was trained for a maximum of 25 epochs with a batch size of 64. In this study, we retrained five deep CNN models: VGG-16, ResNet50, InceptionV3, InceptionResNetV2, and EfficientNetB3. After hyperparameter tuning, all transfer learning models are trained with a batch size of 32 for 30 epochs using the Adam optimizer. The softmax activation function is chosen in the output layer. This work uses an acoustic DNN-based model and RNN-based methods like GRU and LSTM. The RNN models LSTM and GRU are trained using Adam optimization in 30 and 20 epochs, respectively, with a batch size of 32. During the experiment, 10 % of the corpus was used for validation.

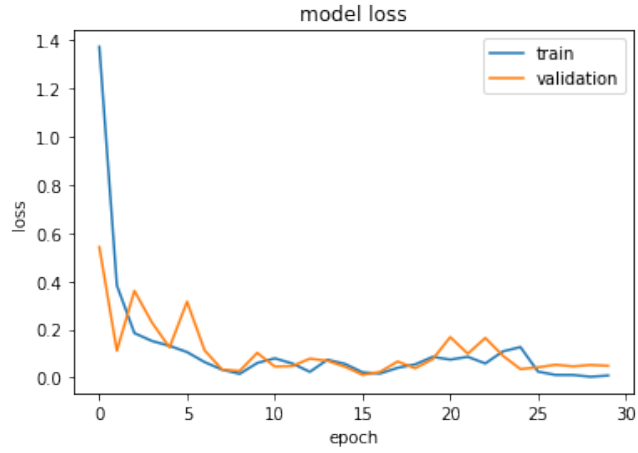
The accuracy and loss for training and validation data in LSTM model is shown in Figure 7. It can be observed from Figure 7 that the model exhibits almost stable but slightly fluctuating curves, and it achieves the highest training and validation accuracy. The accuracy curves for the training data show rapid improvement, reaching around 94 % from epoch 0 to 5, and then stabilizing at a value slightly over 100 % after epoch 25. Similarly, the accuracy curve for the validation data reaches approximately 99 %. As for the loss curves of the training and validation data, the loss quickly decreases to approximately 2 % within the first five epochs, then continues to decrease with slight fluctuations until 25 epochs before stabilizing beyond 25 epochs.

4 RESULTS AND DISCUSSION

The performance metrics with variable slicing length are shown in Figure 8. The 3 s window indeed performed worse than the shorter windows, and 0.5 s is too short



a)



b)

Figure 7. Accuracy and loss for training and validation data in LSTM model

for identifying the bird call. So, we have chosen a slicing length of 1.5 s for the performance evaluation.

The confusion matrix for the Vanilla-DNN, LSTM, VGG-16, and proposed DCNN models for the target dataset comprising two and three species are given in Tables 9, 10, 12 and 13. It is found that the proposed DCNN models outperform the VGG-16 model and the acoustic LSTM model by 10 % and 13 %, respectively. In our experiments, the best-performing DCNN model outperforms all other transfer learning schemes and acoustic models with an accuracy of 75 %.

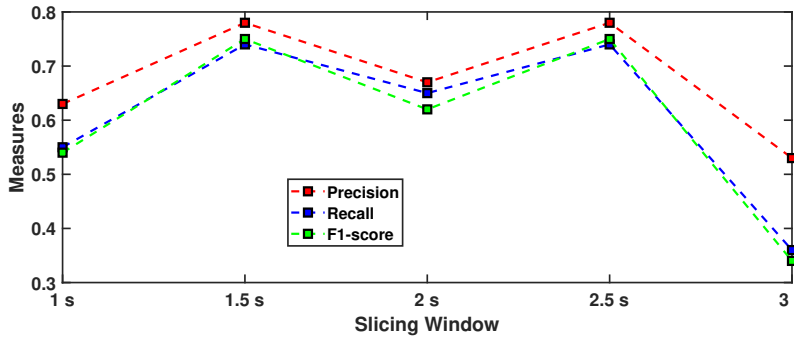


Figure 8. Performance metrics with various slicing window lengths

No.	**Species Name**	**Vanilla-DNN**			**GRU**			**LSTM**		
		P	R	F1	P	R	F1	P	R	F1
1	House Crow	0.76	0.77	0.77	0.90	0.86	0.88	0.86	0.62	0.72
2	Mallard Duck	0.71	0.34	0.46	0.77	0.76	0.76	0.74	0.71	0.73
3	Asian Koel	0.48	0.94	0.64	0.62	0.77	0.69	0.70	0.78	0.73
4	Eurasian Owl	0.36	0.30	0.31	0.52	0.22	0.31	0.71	0.09	0.16
5	House Sparrow	0.38	0.12	0.18	0.80	0.23	0.36	0.56	0.38	0.45
6	Blue Jay	0.62	0.72	0.67	0.37	0.71	0.48	0.44	0.74	0.55
7	Red-wattled Lapwing	0.97	0.58	0.73	0.70	0.52	0.60	0.89	0.57	0.70
8	Grey Go-away	0.51	0.83	0.63	0.48	0.76	0.60	0.50	0.87	0.64
9	Indian Peafowl	0.34	0.52	0.41	0.43	0.46	0.44	0.65	0.63	0.64
10	Western Wood-Pewee	0.96	0.84	0.90	1.00	0.50	0.67	0.93	0.77	0.84
	Macro Average	0.61	0.60	**0.57**	0.66	0.58	**0.58**	0.70	0.62	**0.62**

Table 8. Precision (P), recall (R), and F1 metric of the acoustic DNN/RNN model

The class-wise accuracy of Mallard Duck, Eurasian Owl, House Sparrow, and Indian Peafowl using the Vanilla-DNN approach is less than 50 %. In the proposed DCNN approach, however, all classes report an accuracy greater than 50 %. The proposed DCNN significantly reduced Indian Peafowl, Mallard Duck, and House Sparrow misclassification errors. The class House Sparrow has made significant progress. Compared to the Vanilla-DNN approach, the classification accuracy of three target classes, Grey Go-away, Asian Koel, and Western Wood-Pewee, decreased slightly.

Visualization of feature maps is given in Figure 9. The purpose of visualizing a feature map for a specific input image is to understand which features from the input are being detected or highlighted in these maps, as discussed in [41]. It is generally assumed that feature maps closer to the input layer capture finer details, while those closer to the model's output focus on more generalized characteristics.

	AK	**BJ**	**HC**	**MD**	**GG**	**RL**	**EO**	**IP**	**HS**	**WW**
AK	**65**	0	0	1	3	0	0	0	0	0
BJ	6	**73**	2	6	11	0	1	2	0	0
HC	5	1	**77**	1	13	0	0	2	1	0
MD	5	8	11	**30**	25	0	0	8	1	0
GG	6	13	2	0	**120**	0	0	4	0	0
RL	7	2	0	0	26	**79**	2	7	13	0
EO	9	3	2	2	5	0	**15**	14	5	0
IP	22	0	0	0	2	1	4	**33**	1	0
HS	8	18	7	2	26	1	13	27	**14**	3
WW	2	0	0	0	3	0	7	1	2	**77**

Table 9. Confusion matrix: Vanilla-DNN

	AK	**BJ**	**HC**	**MD**	**GG**	**RL**	**EO**	**IP**	**HS**	**WW**
AK	**54**	6	0	0	3	1	2	2	1	0
BJ	4	**75**	0	3	12	0	0	4	2	1
HC	0	5	**62**	7	22	0	0	2	2	0
MD	0	5	0	**63**	12	2	0	3	3	0
GG	0	16	0	0	**127**	0	0	0	2	0
RL	7	18	0	3	18	**77**	0	3	9	1
EO	1	11	5	4	13	1	**5**	7	8	0
IP	5	1	0	2	15	0	0	**40**	0	0
HS	5	32	5	3	20	6	0	0	**45**	3
WW	2	3	0	0	9	0	0	0	8	**70**

Table 10. Confusion matrix: LSTM

When filters are applied in the initial convolutional layer, it results in multiple variations of the bird call, each emphasizing different attributes. Notably, the highly activated neurons in the first layer across all species strongly indicate their primary role in identifying horizontal edges within the input mel-spectrogram, aiming to detect harmonic components. This observation is in line with our expectations and aligns with our understanding of early-stage feature extraction. To further explore these patterns, we have updated the model to visualize feature maps based on the outputs of other convolutional layers, specifically layers 1, 4, and 8, as depicted in Figure 9. The proposed deep CNN demonstrates its ability to extract more discriminative information from feature maps and effectively preserve critical edges related to multiple overlapping species within the mel-spectrogram. The distinctive spectral patterns of species like Asian Koel, Crow, and Sparrow are clearly discernible. It is worth noting that as we move deeper into the model, the feature maps progressively lose fine-grained detail, as evident from the visualizations. Although it may not be entirely clear from the final image how the model perceived the intricate patterns within the bird call mel-spectrogram, our ability to interpret these deeper feature maps diminishes.

No.	Species Name	VGG-16			ResNet50			InceptionV3		
		P	R	F1	P	R	F1	P	R	F1
1	House Crow	0.76	0.72	0.74	0.70	0.55	0.62	0.77	0.46	0.57
2	Mallard Duck	0.62	0.57	0.60	0.64	0.39	0.48	0.51	0.41	0.45
3	Asian Koel	0.61	0.80	0.70	0.92	0.68	0.78	0.42	0.71	0.53
4	Eurasian Owl	0.67	0.53	0.60	0.80	0.65	0.72	0.32	0.25	0.28
5	House Sparrow	0.71	0.50	0.58	0.80	0.55	0.66	0.70	0.23	0.35
6	Blue Jay	0.51	0.64	0.57	0.49	0.84	0.62	0.45	0.70	0.54
7	Red. Lapwing	0.44	0.50	0.46	0.65	0.43	0.51	0.48	0.43	0.45
8	Grey Go-Away	0.66	0.65	0.66	0.70	0.61	0.65	0.71	0.71	0.71
9	I. Peafowl	0.74	0.89	0.80	0.43	0.94	0.59	0.38	0.90	0.54
10	W. Wood-Pewee	0.82	0.72	0.77	0.48	0.71	0.57	0.86	0.60	0.70
	Macro Average	0.65	0.65	**0.65**	0.66	0.62	**0.62**	0.56	0.54	**0.51**

No.	Species Name	InceptionResNetV2			EfficientNetB3			Proposed DCNN		
		P	R	F1	P	R	F1	P	R	F1
1	House Crow	0.73	0.54	0.62	0.73	0.60	0.65	0.80	0.81	0.80
2	Mallard Duck	0.59	0.60	0.60	0.54	0.67	0.60	0.57	0.57	0.57
3	Asian Koel	0.62	0.75	0.68	0.72	0.74	0.73	1.00	0.81	0.90
4	Eurasian Owl	0.80	0.42	0.55	0.37	0.18	0.24	0.74	0.47	0.58
5	House Sparrow	0.67	0.44	0.53	0.71	0.48	0.57	0.81	0.90	0.85
6	Blue Jay	0.50	0.63	0.56	0.53	0.33	0.40	0.50	0.84	0.63
7	Red. Lapwing	0.58	0.46	0.51	0.51	0.51	0.51	0.93	0.63	0.75
8	Grey Go-Away	0.60	0.67	0.63	0.59	0.63	0.61	0.74	0.76	0.75
9	I. Peafowl	0.40	0.97	0.57	0.40	0.90	0.55	0.73	0.92	0.82
10	W. Wood-Pewee	0.89	0.60	0.71	0.65	0.68	0.67	0.95	0.68	0.80
	Macro Average	0.64	0.61	**0.60**	0.57	0.57	**0.55**	0.78	0.74	**0.75**

Table 11. Performance metrics of the proposed model and transfer learning models

	AK	BJ	HC	MD	GG	RL	EO	IP	HS	WW
AK	**55**	5	2	2	3	2	0	0	0	0
BJ	5	**65**	1	4	8	10	1	3	3	1
HC	2	1	**72**	3	8	7	2	1	3	1
MD	7	2	7	**50**	5	16	0	0	0	1
GG	2	14	6	5	**94**	8	1	1	8	6
RL	12	18	1	10	6	**67**	7	6	4	5
EO	4	5	1	1	3	5	**29**	6	1	0
IP	0	2	2	0	0	0	2	**56**	1	0
HS	3	8	2	3	12	28	1	3	**59**	0
WW	0	6	1	2	3	10	0	0	4	**66**

Table 12. Confusion matrix: VGG-16

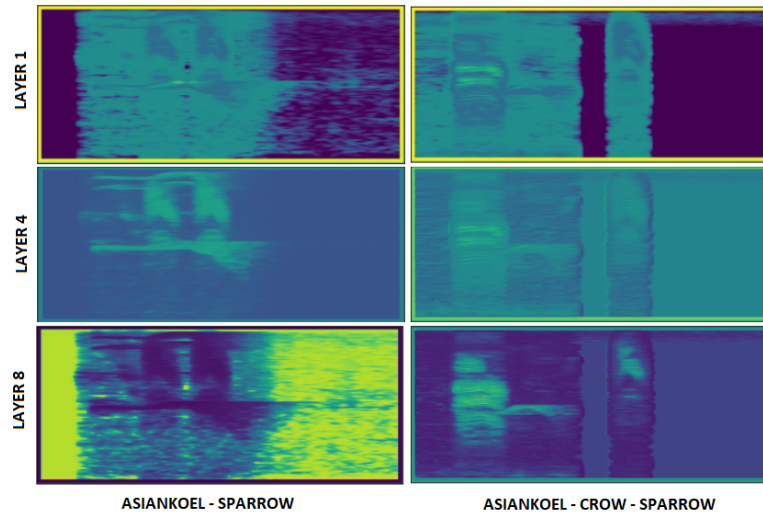


Figure 9. Visualization of feature maps with multiple species: Layer 1 (row 1), Layer 4 (row 2), Layer 8 (row 3) of Asian Koel-Sparrow (column 1), Asian Koel-Crow-Sparrow (column 2), respectively

	AK	**BJ**	**HC**	**MD**	**GG**	**RL**	**EO**	**IP**	**HS**	**WW**
AK	**56**	7	1	2	2	1	0	0	0	0
BJ	0	**85**	4	2	4	2	0	4	0	0
HC	0	8	**81**	2	1	2	0	3	3	0
MD	0	11	5	**50**	16	0	4	0	2	0
GG	0	20	1	4	**111**	0	2	1	6	0
RL	0	11	3	16	7	**86**	2	4	6	1
EO	0	7	4	4	6	0	**26**	5	3	0
IP	0	1	0	2	0	1	0	**58**	0	1
HS	0	6	1	4	0	0	0	0	**107**	1
WW	0	14	1	2	2	0	1	4	5	**63**

Table 13. Confusion matrix: Proposed DCNN

Tables 8 and 11 show the precision, recall, and F1 measure of the experiments. Overall classification scores for VGG-16, ResNet50, InceptionV3, Inception-ResNetV2, EfficientNetB3, Vanilla-DNN, GRU, LSTM, and the proposed DCNN model are 0.65, 0.62, 0.51, 0.60, 0.55, 0.57, 0.58, 0.62, 0.75, respectively. The LSTM framework's macro average precision, recall, and F1 measures are 0.70, 0.62, and 0.62, respectively, while the transfer learning-based VGG-16 model's macro average precision, recall, and F1 measures are 0.65, 0.65, and 0.65, respectively. The metrics reported by the proposed DCNN approach are 0.78, 0.74, and 0.75, respectively. The VGG-16 gives the best performance among the five pre-trained models used. The

average F1 measure for proposed DCNN, VGG-16, and Vanilla-DNN based frameworks are 0.75, 0.65, and 0.57. Compared to the Vanilla-DNN framework based on acoustic cues, there is a significant improvement in visual processing performance. Even in the case of overlapping vocalization, effective pattern learning from visual representation could be a possible cause.

It is worth noting that the proposed DCNN based architecture outperforms the Vanilla-DNN, LSTM, and VGG-16 frameworks. The Vanilla-DNN framework could not perform well for the audio files when overlapping vocalizations are present. By adopting the sequential aggregation approach, DCNN, originally designed for image classification, is adapted and fine-tuned to detect the presence of birds in audio recordings in the proposed work. The majority of the existing frameworks were refined using neural networks pre-trained on ImageNet's "trimmed" Large Scale Visual Recognition Challenge (LSVRC) [42] version, a dataset with almost 1.5 pictures of 1 000 object categories scraped from the web [43]. However, re-training the whole network, not just the final layers, is vital when fine-tuning a network originally intended for image classification.

No.	**Method**	**Precision**	**Recall**	**F1 Metric**
1	Grill and Schlüter [Model 1] [44]	0.50	0.50	0.45
2	Grill and Schlüter [Model 2] [44]	0.51	0.48	0.48
3	Efremova et al. [20]	0.61	0.54	0.53
4	Puget [45] [Transformer]	0.69	0.68	0.67
5	Yang et al. [46] [SENet]	0.65	0.58	0.58
6	Gupta et al. [47] [CNN+GRU]	0.68	0.65	0.67
7	Sequential EfficientNetB3	0.57	0.57	0.55
8	Sequential InceptionV3	0.56	0.54	0.51
9	Sequential InceptionResNetV2	0.64	0.61	0.60
10	Sequential ResNet50	0.66	0.62	0.62
11	Sequential Vanilla-DNN	0.61	0.60	0.57
12	Sequential GRU	0.66	0.58	0.58
13	Sequential LSTM	0.70	0.62	0.62
14	Sequential VGG-16	0.65	0.65	0.65
15	**Proposed Sequential DCNN**	0.78	0.74	**0.75**

Table 14. Performance comparison (Implemented using sequential aggregation and the Xeno-canto dataset)

The performance comparison of various algorithms using our multi-label dataset, assessed in terms of precision, recall, and the F1 metric, is detailed in Table 14. Grill and Schlüter [44] conducted a study comparing two approaches for detecting the presence of bird calls in audio recordings. For model 1 (Global architecture), they reported precision, recall, and an F1 metric of 0.50, 0.50, and 0.45, respectively. Model 2 (Local architecture) achieved slightly improved metrics with precision, recall, and an F1 metric of 0.51, 0.48, and 0.48, respectively. Efremova et al. [20] employed a transfer learning-based ResNet-50 model to evaluate bird call classifica-

tion. With the use of our multi-label dataset, this model achieved an F1 metric of 0.53. Puget [45] proposed an STFT Transformer, where time slices of spectrograms are used as the input patches to the ViT. Another neural network architecture, the SENet, is employed in [46] to enable the network to perform dynamic channel-wise feature re-calibration, which is also mentioned as future work in the paper [48]. The CNN+GRU part of [47] is implemented, and the system gives an F1 metric of 0.67. The F1 metric for our best-performing CNN model using sequential aggregation is 0.75, which is 30 %, 27 %, and 22 % superior to the existing models [44, 20]. It is worth noting that the proposed sequential aggregation strategy shows promise in recognition of bird vocalization in multi-label audio recordings in comparison with the existing models.

5 CONCLUSION

The issue addressed here is identifying multiple bird species from noisy or overlapping raw audio recordings. A DCNN architecture with a sequential aggregation strategy was proposed for the multi-label bird call classification task. Five different transfer learning models and an acoustic DNN/RNN based network were also implemented, and the best outcome in the test data was obtained using our proposed DCNN model. Data augmentation methods like time masking, frequency masking, and time-warping have been proposed to generate additional training data for DCNN learning. The DCNN-based scheme achieves an average F1-metric of 0.75, and it performs better than the transfer learning and acoustic approaches.

REFERENCES

[1] Fagerlund, S.: Automatic Recognition of Bird Species by Their Sound. Master Thesis. Helsinki University of Technology, Finland, 2004.

[2] Jancovic, P.—Köküer, M.: Bird Species Recognition Using Unsupervised Modeling of Individual Vocalization Elements. IEEE/ACM Transactions on Audio, Speech, and Language Processing, Vol. 27, 2019, No. 5, pp. 932–947, doi: 10.1109/TASLP.2019.2904790.

[3] Stowell, D.—Wood, M.—Stylianou, Y.—Glotin, H.: Bird Detection in Audio: A Survey and a Challenge. 2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP), 2016, pp. 1–6, doi: 10.1109/MLSP.2016.7738875.

[4] Gelling, D.: Bird Song Recognition Using GMMs and HMMs. Master Thesis. University of Sheffield, 2001.

[5] Thakur, A.—Abrol, V.—Sharma, P.—Rajan, P.: Local Compressed Convex Spectral Embedding for Bird Species Identification. The Journal of the Acoustical Society of America, Vol. 143, 2018, No. 6, pp. 3819–3828, doi: 10.1121/1.5042241.

[6] Stowell, D.—Wood, M. D.—Pamuła, H.—Stylianou, Y.—Glotin, H.: Automatic Acoustic Detection of Birds Through Deep Learning: The First Bird Au-

dio Detection Challenge. Methods in Ecology and Evolution, Vol. 10, 2019, No. 3, pp. 368–380, doi: 10.1111/2041-210X.13103.

[7] Potamitis, I.—Ntalampiras, S.—Jahn, O.—Riede, K.: Automatic Bird Sound Detection in Long Real-Field Recordings: Applications and Tools. Applied Acoustics, Vol. 80, 2014, pp. 1–9, doi: 10.1016/j.apacoust.2014.01.001.

[8] Thakur, A.—Abrol, V.—Sharma, P.—Rajan, P.: Compressed Convex Spectral Embedding for Bird Species Classification. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 261–265, doi: 10.1109/ICASSP.2018.8461814.

[9] Noumida, A.—Rajan, R.: Deep Learning-Based Automatic Bird Species Identification from Isolated Recordings. 8th International Conference on Smart Computing and Communications (ICSCC), 2021, pp. 252–256, doi: 10.1109/ICSCC51209.2021.9528234.

[10] Thakur, A.—Abrol, V.—Sharma, P.—Rajan, P.: Deep Convex Representations: Feature Representations for Bioacoustics Classification. Proceedings of Interspeech 2018, 2018, pp. 2127–2131, doi: 10.21437/Interspeech.2018-1705.

[11] Rajan, R.—Johnson, J.—Abdul Kareem, N.: Bird Call Classification Using DNN-Based Acoustic Modelling. Circuits, Systems, and Signal Processing, Vol. 41, 2022, No. 5, pp. 2669–2680, doi: 10.1007/s00034-021-01896-2.

[12] Briggs, F.—Lakshminarayanan, B.—Neal, L.—Fern, X. Z.—Raich, R.—Hadley, S. J. K.—Hadley, A. S.—Betts, M. G.: Acoustic Classification of Multiple Simultaneous Bird Species: A Multi Instance Multi-Label Approach. The Journal of the Acoustical Society of America, Vol. 131, 2012, No. 6, pp. 4640–4650, doi: 10.1121/1.4707424.

[13] Zhang, L.—Towsey, M.—Xie, J.—Zhang, J.—Roe, P.: Using Multi-Label Classification for Acoustic Pattern Detection and Assisting Bird Species Surveys. Applied Acoustics, Vol. 110, 2016, pp. 91–98, doi: 10.1016/j.apacoust.2016.03.027.

[14] Noumida, A.—Rajan, R.: Multi-Label Bird Species Classification from Audio Recordings Using Attention Framework. Applied Acoustics, Vol. 197, 2022, Art. No. 108901, doi: 10.1016/j.apacoust.2022.108901.

[15] Noumida, A.—Mukund, R.—Nair, N. M.—Rajan, R.: Multi-Label Bird Species Classification Using Ensemble of Pre-Trained Networks. International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS), IEEE, 2023, pp. 644–649, doi: 10.1109/ICISCoIS56541.2023.10100519.

[16] Noumida, A.—Rajan, R.: Multi-Label Bird Species Classification Using Hierarchical Attention Framework. 2022 IEEE 19th India Council International Conference (INDICON), 2022, pp. 1–6, doi: 10.1109/INDICON56171.2022.10039791.

[17] Rajan, R.—Noumida, A.: Multi-Label Bird Species Classification Using Transfer Learning. International Conference on Communication, Control and Information Sciences (ICCISc), IEEE, Vol. 1, 2021, pp. 1–5, doi: 10.1109/ICCISc52257.2021.9484858.

[18] Sprengel, E.—Jaggi, M.—Kilcher, Y.—Hofmann, T.: Audio Based Bird Species Identification Using Deep Learning Techniques. In: Balog, K., Cappellato, L., Ferro, N., Macdonald, C. (Eds.): Working Notes of CLEF 2016 (Conference and Labs of the Evaluation Forum). CEUR Workshop Proceedings, Vol. 1609, 2016,

pp. 547–559, https://ceur-ws.org/Vol-1609/16090547.pdf.

[19] Kahl, S.—Wilhelm-Stein, T.—Hussein, H.—Klinck, H.—Kowerko, D.—Ritter, M.—Eibl, M.: Large-Scale Bird Sound Classification Using Convolutional Neural Networks. In: Cappellato, L., Ferro, N., Goeuriot, L., Mandl, T. (Eds.): Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum. CEUR Workshop Proceedings, Vol. 1866, 2017, https://ceur-ws.org/Vol-1866/paper_143.pdf.

[20] Efremova, D. B.—Sankupellay, M.—Konovalov, D. A.: Data-Efficient Classification of Birdcall Through Convolutional Neural Networks Transfer Learning. 2019 Digital Image Computing: Techniques and Applications (DICTA), 2019, pp. 1–8, doi: 10.1109/DICTA47822.2019.8946016.

[21] Seppänan, J.: Computational Models for Musical Meter Recognition. Master Thesis. Tampere University of Technology, Tampere, Finland, 2001.

[22] Richard, G.—Sundaram, S.—Narayanan, S.: An Overview on Perceptually Motivated Audio Indexing and Classification. Proceedings of the IEEE, Vol. 101, 2013, No. 9, pp. 1939–1954, doi: 10.1109/JPROC.2013.2251591.

[23] Gruber, N.—Jockisch, A.: Are GRU Cells More Specific and LSTM Cells More Sensitive in Motive Classification of Text? Frontiers in Artificial Intelligence, Vol. 3, 2020, Art. No. 40, doi: 10.3389/frai.2020.00040.

[24] Tang, C. P.—Chui, K. L.—Yu, Y. K.—Zeng, Z.—Wong, K. H.: Music Genre Classification Using a Hierarchical Long Short Term Memory (LSTM) Model. In: Jiang, X., Chen, Z., Chen, G. (Eds.): Third International Workshop on Pattern Recognition. SPIE, Proceedings of SPIE, Vol. 10828, 2018, pp. 334–340, doi: 10.1117/12.2501763.

[25] Olah, C.: Understanding LSTM Networks. 2015, https://colah.github.io/posts/2015-08-Understanding-LSTMs/ [accessed 25-Feb-2020].

[26] Géron, A.: Hands-on Machine Learning with Scikit-Learn and Tensorflow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'reilly Media, Inc., 2017.

[27] Chung, J.—Gulcehre, C.—Cho, K.—Bengio, Y.: Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. CoRR, 2014, doi: 10.48550/arXiv.1412.3555.

[28] Cho, K.—van Merrienboer, B.—Gulcehre, C.—Bahdanau, D.—Bougares, F.—Schwenk, H.—Bengio, Y.: Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. CoRR, 2014, doi: 10.48550/arXiv.1406.1078.

[29] Sukhavasi, M.—Adapa, S.: Music Theme Recognition Using CNN and Self-Attention. CoRR, 2019, doi: 10.48550/arXiv.1911.07041.

[30] Ghosal, D.—Kolekar, M. H.: Music Genre Recognition Using Deep Neural Networks and Transfer Learning. Proceedings of Interspeech 2018, 2018, pp. 2087–2091, doi: 10.21437/Interspeech.2018-2045.

[31] LeCun, Y.—Bengio, Y.—Hinton, G.: Deep Learning. Nature, Vol. 521, 2015, No. 7553, pp. 436–444, doi: 10.1038/nature14539.

[32] Salamon, J.—Bello, J. P.: Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification. IEEE Signal Processing Letters,

Vol. 24, 2017, No. 3, pp. 279–283, doi: 10.1109/LSP.2017.2657381.

[33] Park, D. S.—Chan, W.—Zhang, Y.—Chiu, C. C.—Zoph, B.—Cubuk, E. D.—Le, Q. V.: SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. 2019, pp. 2613–2617, doi: 10.21437/Interspeech.2019-2680.

[34] Fritzler, A.—Koitka, S.—Friedrich, C. M.: Recognizing Bird Species in Audio Files Using Transfer Learning FHDO Biomedical Computer Science Group (BCSG). Vol. 1866, 2017, Art. No. 169, https://ceur-ws.org/Vol-1866/paper_169.pdf.

[35] Tan, M.—Le, Q. V.: EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In: Chaudhuri, K., Salakhutdinov, R. (Eds.): Proceedings of the 36th International Conference on Machine Learning (ICML). Proceedings of Machine Learning Research (PMLR), Vol. 97, 2020, pp. 6105–6114, http://proceedings.mlr.press/v97/tan19a/tan19a.pdf.

[36] Vellinga, W. P.—Planqué, R.: The Xeno-Canto Collection and Its Relation to Sound Recognition and Classification. In: Cappellato, L., Ferro, N., Jones, G. J. F., San Juan, E. (Eds.): Working Notes of CLEF 2015 - Conference and Labs of the Evaluation Forum. CEUR Workshop Proceedings, Vol. 1391, 2015, https://ceur-ws.org/Vol-1391/166-CR.pdf.

[37] Brandes, T. S.: Automated Sound Recording and Analysis Techniques for Bird Surveys and Conservation. Bird Conservation International, Vol. 18, 2008, No. S1, pp. 163–173, doi: 10.1017/S0959270908000415.

[38] Duan, S.—Towsey, M.—Zhang, J.—Truskinger, A.—Wimmer, J.—Roe, P.: Acoustic Component Detection for Automatic Species Recognition in Environmental Monitoring. 2011 Seventh International Conference on Intelligent Sensors, Sensor Networks and Information Processing, 2011, pp. 514–519, doi: 10.1109/ISSNIP.2011.6146597.

[39] Kaya, M.—Bilge, H. Ş.: Deep Metric Learning: A Survey. Symmetry, Vol. 11, 2019, No. 9, Art. No. 1066, doi: 10.3390/sym11091066.

[40] Liu, C.—Feng, L.—Liu, G.—Wang, H.—Liu, S.: Bottom-Up Broadcast Neural Network for Music Genre Classification. Multimedia Tools and Applications, Vol. 80, 2021, No. 5, pp. 7313–7331, doi: 10.1007/s11042-020-09643-6.

[41] Zeiler, M. D.—Fergus, R.: Visualizing and Understanding Convolutional Networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (Eds.): Computer Vision – ECCV 2014. Springer, Cham, Lecture Notes in Computer Science, Vol. 8689, 2013, pp. 818–833, doi: 10.1007/978-3-319-10590-1_53.

[42] Russakovsky, O.—Deng, J.—Su, H.—Krause, J.—Satheesh, S.—Ma, S.—Huang, Z.—Karpathy, A.—Khosla, A.—Bernstein, M.—Berg, A. C.—Li, F. F.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision, Vol. 115, 2015, pp. 211–252, doi: 10.1007/s11263-015-0816-y.

[43] Lasseck, M.: Acoustic Bird Detection with Deep Convolutional Neural Networks. Detection and Classification of Acoustic Scenes and Events 2018, 2015, pp. 143–147, https://dcase.community/documents/workshop2018/proceedings/DCASE2018Workshop_Lasseck_134.pdf.

[44] Grill, T.—Schlüter, J.: Two Convolutional Neural Networks for Bird Detection

in Audio Signals. 2017 25th European Signal Processing Conference (EUSIPCO), IEEE, 2017, pp. 1764–1768, doi: 10.23919/EUSIPCO.2017.8081512.

[45] Puget, J. F.: STFT Transformers for Bird Song Recognition. In: Faggioli, G., Ferro, N., Joly, A., Maistro, M., Piroi, F. (Eds.): Proceedings of the Working Notes of CLEF 2021 – Conference and Labs of the Evaluation Forum. CEUR Workshop Proceedings, Vol. 2936, 2021, pp. 1609–1616, https://ceur-ws.org/Vol-2936/paper-137.pdf.

[46] Yang, F.—Jiang, Y.—Xu, Y.: Design of Bird Sound Recognition Model Based on Lightweight. IEEE Access, Vol. 10, 2022, pp. 85189–85198, doi: 10.1109/ACCESS.2022.3198104.

[47] Gupta, G.—Kshirsagar, M.—Zhong, M.—Gholami, S.—Ferres, J. L.: Comparing Recurrent Convolutional Neural Networks for Large Scale Bird Species Classification. Scientific Reports, Vol. 11, 2021, No. 1, Art. No. 17085, doi: 10.1038/s41598-021-96446-w.

[48] Koh, C. Y.—Chang, J. Y.—Tai, C. L.—Huang, D. Y.—Hsieh, H. H.—Liu, Y. W.: Bird Sound Classification Using Convolutional Neural Networks. In: Cappellato, L., Ferro, N., Losada, D. E., Müller, H. (Eds.): Working Notes of CLEF 2019 – Conference and Labs of the Evaluation Forum. CEUR Workshop Proceedings, Vol. 2380, 2019, https://ceur-ws.org/Vol-2380/paper_68.pdf.

Noumida Abdul Kareem received her B.Tech. in electronics and communication from the College of Engineering, Chengannur (Cochin University of Science and Technology, India) in 2018, and M.Tech. in signal processing from the College of Engineering, Trivandrum in 2021. She is currently pursuing her Ph.D. degree with the College of Engineering, Trivandrum. Her research areas are audio signal processing and bioacoustics.

Rajeev Rajan received his B.Tech. in electronics and communication from the College of Engineering, Adoor (Cochin University of Science and Technology, India) in 2000, M.Tech. in applied electronics and instrumentation from the College of Engineering, Trivandrum in 2004, and Ph.D. from the Department of Computer Science and Engineering, Indian Institute of Technology, Madras, Chennai in 2017. His research areas are speech and music signal processing, image processing.