Computing and Informatics, Vol. 42, 2023, 781–804, doi: 10.31577/cai_2023_4_781

PARAMETERIZED REACHABILITY GRAPH FOR SOFTWARE MODEL CHECKING BASED ON PDNET

Xiangyu Jia, Shuo Li*

Department of Computer Science and Technology
Tongji University
Shanghai 201804, China
e-mail: jiaxy1999@163.com, lishuo2017@tongji.edu.cn

Abstract. Model checking is a software automation verification technique. However, the complex execution process of concurrent software systems and the exhaustive search of state space make the model-checking technique limited by the state-explosion problem in real applications. Due to the uncertain input information (called system parameterization) in concurrent software systems, the state-explosion problem in model checking is exacerbated. To address the problem that reachability graphs of Petri net are difficult to construct and cannot be explored exhaustively due to system parameterization, this paper introduces parameterized variables into the program dependence net (a concurrent program model). Then, it proposes a parameterized reachability graph generation algorithm, including decision algorithms for verifying the properties. We implement LTL-χ verification based on parameterized reachability graphs and solve the problem of difficulty constructing reachability graphs caused by uncertain inputs.

Keywords: Model checking, PDNet, parameterized reachability graph

Mathematics Subject Classification 2010: 68-Q60

* Corresponding author

1 INTRODUCTION

With the rapid development of information technology, software systems have become increasingly large and complex, and the number of defects in software systems has increased dramatically. It has become challenging to verify software programs solely by manual inspection [1], and it is urgent to develop automated verification methods to solve this problem to help programmers quickly discover defects in software systems [2, 3, 4, 5]. Formal verification methods have received increasing attention in existing research.

Formal verification techniques include two main approaches, i.e., theorem proving and model checking [6, 7]. Theorem proving can represent the system and properties to be verified as logical formulas in a suitable logical system and then use a theorem prover to prove whether the properties are satisfied in the system [8, 9]. The advantage of theorem proving is that it can be applied to most systems, including infinite-state systems. Its disadvantage is that it is not highly automatic and requires much manual intervention while proving. However, theorem proving does not provide relevant diagnostic information if the formula is falsified. Model checking is one of the most promising automatic verification methods for concurrent software systems [10, 11], and it is an algorithmic approach to verify whether a given model satisfies a particular property expressed by a temporal logic formula using a state space search [7, 12]. For finite state systems, this problem is decidable, i.e., it can be determined automatically in finite time by using a computer program [13, 14, 15, 16]. It verifies the specification through an exhaustive state space enumeration, aiming to achieve higher reliability, correctness, and satisfiability. The advantage of model-checking techniques is that they are highly automated and do not require extensive logic knowledge. When the system does not satisfy a certain property, the model-checking tool returns a counterexample. The interpretation of the counterexample gives the reason why the property does not hold and provides important clues for the correction. There have been many powerful model checkers, such as SPIN [17] and NuSMV [18]. In addition, many reduction techniques have been developed to alleviate the state-explosion problem, such as symbolic model checking, partial order reduction, and symmetry reduction [19, 20].

Petri nets are an important formal model, and they are powerful in describing the internal execution and external interactions of concurrent systems. In contrast to other formal models such as automaton and communication sequential process (CSP), Petri nets can represent true-concurrency rather than interleaving semantics, and they can provide a compact and comprehensive description of control, synchronization, and data operations [21, 22, 23, 24, 25, 26, 27, 28, 29]. However, since the exponential growth of the state space with the increase of the actual software system size, in many cases, the reachability graph analysis method is not feasibly caused by the calculation complexity. On the other hand, since the reachability graph is calculated based on the initial marking, if the initial input parameters are uncertain, a completely different reachability graph may have to be calculated for each assignment of the input parameters. The uncertainty of the input may

not generate a reachability graph, resulting in the inability to analyze the properties.

To solve the challenge caused by the uncertain input, this paper proposes a parameterized reachability graph for software model checking based on Program Dependence Net (PDNet) [30]. The main contributions of this paper are as follows:

1. Parameterized reachability graph based on PDNet is proposed by introducing the definition of parameterized variables in PDNet. We define the corresponding occurrence rules and make it possible to generate parameterized reachability graphs even for PDNet with uncertain inputs.
2. The generation algorithm for the parameterized reachability graphs is proposed. It classifies markings using parameterized marking and then uses these parameterized reachability graphs to perform a determination for model checking.
3. We implemented the parameterized reachability graph generation algorithm on DAMER, a concurrent program model checking tool based on PDNet, to enhance the ability of DAMER to handle uncertain input parameters.

Section 2 presents some related works. Section 3 introduces the definition of PDNet based on multisets and Color Petri Net (CPN). Section 4 proposes the definition with parameterized variables, including the corresponding algorithm for generating parameterized reachability graphs. Section 5 verifies the effectiveness of our algorithms through comparative experiments. Section 6 concludes the paper and gives some following works.

2 RELATED WORKS

Model checking is a technique used to automatically verify the correct behavioral properties of a computer system. The basic approach is to use a state transition graph to represent the model of the system under test and to describe the correct behavioral properties of the computer system using computation tree logic (CTL), and linear temporal logic (LTL). Correct behavioral properties of the computer system. The main bottleneck of model checking in practical applications is the state explosion problem. In 1987, McMillan adopted a symbolic approach to representing a state transition graph that allowed him to check large-scale systems [31]. This method is based on Bryant's ordered bifurcation decision diagram (OBDD) [32], which is more concise than the conjunction or disjunction normal form. His team also developed an efficient OBDD algorithm and a symbolic model checking system SMV [33]. Symbolic methods are suitable for hardware system verification with strong structured features and have achieved many successful cases. Still, software systems are less structured than hardware, and concurrent software is asynchronous, so software system verification poses some problems for model checking. Partial order reduction has made great progress in suppressing the state space explosion of software systems [34, 35, 36], and the technique is based on the independence between concurrent events to approximate the state space of a model by reducing

the number of interleaved sequences. The partial order reduction technique treats all independent intertwined executions on the transition relations between states as a set. It selects its subsets to reduce its state space, with significant strategies such as Peled's ample sets [35], Valmari's stubborn sets [36], Godefroid's solid and sleeping sets [37], etc. Although symbolic methods and partial order reduction techniques greatly increase the size of verifiable systems, many practical applications are too large to handle the problem size caused by uncertain inputs; therefore, it becomes important to find new techniques to enhance verification in combination with symbolic methods. Petri nets not only have a rich theoretical foundation but also have graphical features, which are more intuitive and easier to understand than algebraic descriptions in textual form. Reachability graphs are the main analysis method for Petri Net models. Because the classical reachability graph cannot handle the model of the checked system that contains parameters or uncertain inputs, it makes the model properties of the checked system becomes very difficult. Usually, parameterized reachability graph (PRG) and symbolic reachability graph (SRG) is used to solve this problem.

The core idea of PRG is to simplify the reachability graph using state classification, and the representation of the state is parameterized. The state classification in the parameterized approach will depend on whether certain specific conditions hold. The literature [38] proposes a method for constructing parameterized reachability graphs based on Petri nets, which defines two kinds of partial order relations for parameterized state marking: $\supseteq$ and $>$. It parametrizes the marking can represent all reachable marking of the verified system and defines the execution of all instantiation procedures; [39] defines a transition implementation rule for PRG, which first calculates the parametrized marking of each place in the reachability graph based on the incoming arcs and outgoing arcs of that place, and splits the marking if the parametrized marking cannot represent the same transition; If the parameterized marking is larger than one of its ancestors, infinite branching should be avoided. The relevant properties of the system are verified based on the enabled and occurrence rules.

Since this approach uses integers to represent the minimum number of tokens in a place, it results in its inability to fully express the information in a parallel program when faced with a parallel program. It requires the definition of new symbolic tokens for description.

The main idea of SRG [40] is to use the inherent symmetry of the system to obtain a compressed representation of the reachable states, which is also a simplified representation of the Well-Formed Colored Petri Net (WN) reachability graph. The SRG simplifies the state representation based on the symmetry of WN by introducing a color function syntax definition to reduce the state space. The SRG is constructed directly by using symbolic marking to represent the equivalence classes in the WN state space, and by introducing the canonical representation of symbolic marking and the enabled and occurrence rules, the SRG is constructed by the same algorithm as the regular reachability graph, except that the SRG uses canonical symbolic marking instead of initial marking and the ordinary enabled and occurrence rules.

Based on SRG and WN theory, [41] defines Stochastic Well-Formed Colored Nets (SWN), which introduce syntactic restriction rules in SWN to reduce the complexity of Markov performance evaluation using SRG. SWN allows to represent of any color function in a structured form so that any unrestricted high-level semantic net can be transformed into a canonical net; [42] defines Extend Symbolic Reachability Graph (ESRG), which simplifies the state space of the checked system by exploiting the partial symmetry in the WN net, and the model analysis and simulation algorithms automatically exploit the model symmetry to improve their efficiency.

It is worth pointing out that the reduction of the SRG approach for reachability graphs strongly relies on the symmetry of the model itself. the more equivalent behaviors between model objects, the more symbolic marking in the same equivalence class, and thus the higher the state compression rate of the original state reachability graph. SRG does not provide significant gains when asymmetric actions appear in the behavior description.

The above methods alleviate the problem of difficulty in constructing the reachability graphs of the Petri net model due to the system parameterization, which leads to the inability of space state exploration, and thus has some limitations in model checking. Based on the analysis of existing reachability graph methods, we propose a new reachability graph construction method using parameterized marking to solve the problem of difficult generation of reachability graph for Petri net caused by uncertain input.

3 PDNET WITH PARAMETERIZED VARIABLES

3.1 Introduction of PDNet

PDNet is our previously proposed model based on CPN, which combines the control-flow structure and dependencies. To define PDNet, we first introduce the definition of multiset and CPN.

Definition 1 (Multiset)**.** Let S be a non-empty set. A multiset $ms : S \to N$ on S is a function that maps each element to a non-negative integer. S_{MS} is the set of all multisets over S. We use $+$ and $-$ for the sum and difference of two multisets. $=$, $>$, $<$, $\geq$, $\leq$ are comparisons of multisets, which are defined in the standard way.

Also, we give some symbolic terms for the following definitions: $BOOL = \{false, true\}$ is the set of Boolean predicates with standard logical operations; $EXPR$ is the set of expressions; $Type[e]$ is the type of an expression $e \in EXPR$, i.e., the type of the value obtained when evaluating e; $Var(e)$ is the set of all variables in an expression e; $EXPR_V$ for a variable set V is the set of expressions $e \in EXPR$ such that $Var(e) \in V$. $Type[v]$ is the type of a variable v.

Definition 2 (Colored Petri Nets)**.** CPN is defined by a 9-tuple,

$$N ::= (\Sigma, V, P, T, F, C, G, E, I),$$

where:

1. Σ is a finite non-empty set of types called color sets;
2. V is a finite set with type variables, $\forall v \in V$, there is $Type[v] \in \Sigma$;
3. P is a finite set of places;
4. T is a finite set of transitions and $T \cap P = \emptyset$;
5. $F \subseteq (P \times T) \cup (T \times P)$ is a finite set of directed arcs;
6. $C : P \to \Sigma$ is a color set function that assigns the color set $C(p)$ belonging to the type set Σ to each place p;
7. $G : T \to EXPR_V$ is a guard function, that assigns an expression $G(t)$ to each transition t, $\forall t \in T : (Type[G(t)] \in BOOL) \wedge (Type[Var(G(t))] \subseteq \Sigma)$;
8. $E : F \to EXPR_V$ is a function, that assigns an arc expression $E(f)$ to each arc f, $\forall f \in F : (Type[E(f)] = C(p(f))_{MS}) \wedge (Type[Var(E(f))] \subseteq \Sigma)$, where $p(f)$ is the place connected arc f;
9. $I : P \to EXPR_\emptyset$ is an initialization function, that assigns an initialization expression $I(p)$ to each place p, $\forall p \in P : (Type[I(p)] = C(p)_{MS}) \wedge (Var(I(p)) = \emptyset)$.

PDNet is also a 9-tuple, which differs from CPN in P and F.

1. P is divided into three subsets, i.e., $P = P_c \cup P_v \cup P_f$. Concretely, P_c is a subset of control places, P_v is a subset of variable places, and P_f is the subset of execution places.
2. F is divided into three subsets, i.e., $F = F_c \cup F_{rw} \cup F_f$, Concretely, F_c is a subset of control arcs, F_{rw} is a subset of read-write arcs, and F_f is a subset of execution arcs.

Except for the two differences, the other definitions and constraints of PDNet are consistent with CPN, and in the following definitions, we give some basic concepts of PDNet.

Definition 3 (Basic concepts in PDNet)**.**

1. $M : P \to EXPR_\emptyset$ is a marking function that assigns an expression $M(p)$ to each place p, $\forall p \in P : Type[M(p)] = C(p)_{MS} \wedge (Var(M(p)) = \emptyset)$; for convenience, the marking of N is denoted by M or $\bar{M}$ with subscript, and in particular, M_0 represents the initial marking $\forall p \in P : M_0(p) = I(p)$;
2. $Var(t) \subseteq V$ is the variable set of a transition t. It consists of the variables appearing in the expression $G(t)$ and arc expressions of all arcs connected to the transition t;
3. $B : V \to CON$ is a binding function that assigns a constant value $B(v)$ to each variable v. $B[t]$ is the set of all bindings of a transition t, that maps $V \in Var(t)$ to a constant value, and $b \in B[t]$ is a binding of a transition t;
4. A binding element (t, b) is a pair where $t \in T$ and $b \in B[t]$, $B[t]$ is a set of all binding elements of a transition t.

3.2 Parameterized Variables

Formally, $e\langle b\rangle$ represents the evaluation result of expression e in binding b by assigning a constant to each variable $v \in Var(e)$ through binding b. Therefore, under the binding element (t, b), the evaluation result of $G(t)$ (or $E(f)$) is represented by $G(t)\langle b\rangle$ (or $E(f)\langle b\rangle$), where f is the arc connected to the transition t. Here, we specifically use v_s to denote parameterized variables and V_s to denote the set of parameterized variables, where $v_s \in V_S, V_S \subseteq V$.

Definition 4 (Parameterized variables for PDNet)**.**

1. e_s: assuming that the assignment operator to the parameterized variable v_s is $v_s := \omega$, e_s is an expression obtained by computing ω based on the current execution state and is any expression involving a unitary or binary operator with symbols and specific values;
2. $EXPR_s$: any finite set of expressions involving variables $v \in V$ and constants $o \in CON$ for unitary or binary operators, $e_s \in EXPR_s$;
3. σ: denotes the symbolic state, a mapping from a variable to a symbolic expression e_s, denoted $\sigma : v_s \mapsto e_s$, i.e. $\sigma(v_s) = e_s$;
4. $SS : V_s \mapsto EXPR_s$, the set of symbolic storage functions $\sigma \in SS$.

In particular, since the parameterized variables do not have a definite value, making it difficult to determine the relationship between their value intervals and the constraints, we also need to define the function $SAT()$, whose input is a string of first-order formulas without quantifiers, which uses the constraint solver [43] to solve for the existence of a solution to the input quantifier-free first-order formulas, with the output being *true* or $false$; if a solution exists for $PC \wedge \sigma(G(t))$, then it is written as $SAT(PC \wedge \sigma(G(t))) = true$.

In the existing PDNet, P is divided into three subsets, $P = P_c \cup P_v \cup P_f$, where P_c is a subset of the control place, P_v is a subset of the variable place, and P_f is a subset of the execution place. We refer to the structure of the original variable place P_v to add a new class of parameterized variable place, denoted as P_s. That is, P is divided into four subsets $P = P_c \cup P_v \cup P_f \cup P_s$, where P_s is defined.

Definition 5 (Parameterized variable place in PDNet)**.** The parameterized variable place P_s is used to store the unassigned variables v_s. The parameterized variable place consists of a triple $\langle \sigma, PC, id\rangle$, where:

1. σ is a symbolic state representing the mapping from variables to parameterized expressions e_s;
2. PC is a quantifier-free first-order formula consisting of the expressions in $G(t)$ on the path and the truth-value selection of the expressions connected to describe the path constraints;
3. id is a unique marking of the P_s place.

where the initial value of the symbolic state σ is *null* and the initial value of the path constraint PC is *true*.

Definition 6 (Parameterized marking and parameterized binding)**.** Parameterized marking $M_s\colon P \to EXPR_{\emptyset}$ is a parameterized marking function that specifies an expression $M_s(p)$ for each variable place p:

$$\forall p \in P : Type[M_s(p)] = C(p)_{MS} \wedge (Var(M_s(p)) = \emptyset).$$

For simplicity of representation, the parameterized marking of N is represented by M_s or M_s with subscript when $M_s(p)$ is present.

For a PDNet N whose variables are all non-parameterized, the marking function is $M : (P \setminus P_s) \to EXPR_{\emptyset}$, specifying an expression $M(p_v)$ for each non-parameterized variable banked by p_v:

$$\forall p_v \in (P \setminus P_s : Type[M(p_v)] = C(p_v)_{MS} \wedge (Var(M(p_v)) = \emptyset).$$

For convenience, the marking of N whose variables are all non-parameterized is denoted by M or M with subscripts. At the same time, we cannot determine a fixed binding element (t, b) for the transition t associated with the parameterized variable place by P_s; since the values of the parameterized variables represented by the parameterized variable place by P_s are jointly represented by σ and PC, there does not exist a specific value to take, and we can consider the range of values as a concatenation of one or more intervals; Since it costs more time and space to solve the value interval of each variable using the constraint solver, we do not directly calculate the value interval of the variables, but determine whether the transition can be enabled under the parameterized marking M_s by analyzing the relationship between σ and PC; define the parameterized binding element (t, σ, PC), where $t \in T, \sigma \in SS$; if the symbolic states and path constraints recorded in the parameterized binding element are covered by $M_s(p)$ after analysis, it is written as $E(p, t)\langle\sigma, PC\rangle \leq M_s(p)$.

Definition 7 (Parameterized enabled and occurrence rules)**.** Let N be a PDNet, (t, b) be a binding element on N, M be a marking on N, and the binding element (t, b) is enabled under the marking M, denoted $M[(t, b)\rangle$, when and only when:

1. $G(t)\langle b\rangle = true$;
2. $\forall p \in {}^\bullet t : E(p, t)\langle b\rangle \leq M(p)$;

   When (t, b) is enabled under M, triggering the transition t leads to the generation of a new marking M_1, denoted as $M[(t, b)\rangle M_1$, such that:
3. $\forall p \in P : M_1(p) = M(p) - E(p, t)\langle b\rangle + E(t, p)\langle b\rangle$.

For parameterized variables, when (t, σ, PC) is enabled under M_s, it may lead to the generation of a new marking M_{s1}, denoted as $M_s[(t, \sigma, PC)\rangle M_{s1}$, when and only when:

1. $SAT(PC \wedge \sigma(G(t))) = true$;
2. $\forall p \in^{\bullet} t : E(p,t)\langle \sigma, PC \rangle \leq M_s(p)$;
3. $\forall p \in P : M_{s1}(p) = M_s(p) - E(p,t)\langle \sigma, PC \rangle + E(t,p)\langle \sigma, PC \rangle$.

The intuition of this rule is to update the path constraint and symbolic state stored in each token, $PC = PC \wedge \sigma(G(t))$, and not to update if $G(t)$ does not contain symbolic variables, see Algorithm 1 for the specific update algorithm. In particular, the two operation cases that we may encounter in the process of updating the symbolic state σ information in Algorithm 1 to define the variable v_s are the input operation and the assignment operation, where the input operation is an external input to the parameterized variable v_s in the form $v_s := sym\_input()$ and the assignment operation is an assignment of a value or expression to the parameterized variable v_s in the form $v_s := \omega$. The symbolic states and path constraints in the parameterized variable place are updated continuously as the parameterized binding elements are enabled and occur.

Algorithm 1 Parameterized variable place information update

Step 1. Determine whether σ, PC in the parameterized variables v_s satisfy the conditions in $G(t)$: $SAT(PC \wedge \sigma(G(t))) = true$;

Step 2. Update the value stored in the path constraint PC.
If $SAT(PC \wedge \neg\sigma(G(t))) = false$ or $G(t)$ does not contain constraints associated with the parameterized variables v_s **Then**
Not updating the contents stored in the PC;
Else
$PC' = PC \wedge \sigma(G(t))$;

Step 3. Update symbol status σ.
If Performing input operations on variables v_s **Then**
Update the mapping σ in v_s to: $v_s \rightarrow v_{si}$, where the initial value of i is 0 and the value of i takes increasing values with the update of the input mapping;
If Assign a value to the variable v_s in the form $a := \omega$ **Then**
Substitute the existing mapping σ in v_s into the formula ω to calculate the new mapping expression, and update σ with the new mapping expression.

The following example shows the update process of symbolic state σ and path constraint PC in the parameterized place, as detailed in Figure 1.

Definition 8 (Occurrence sequence of PDNet)**.** Let N be a PDNet, M_0 be the initial marking of N, and (t,b) be the binding elements of N. The sequence of occurrences in N can be defined by induction:

1. $M_0[\varepsilon\rangle M_0$,($\varepsilon$ is a null sequence);
2. $M_0[\omega\rangle M_1 \wedge M_1[(t,b)\rangle M_2 : M_0[\omega(t,b)\rangle M_2$.

The sequence of occurrences ω in N is maximal when and only when:

1. ω is infinite, e.g., (t_1, b_1), (t_2, b_2), ... or
2. $M_0[\omega\rangle M_1 \wedge \forall t \in T, \nexists (t, b) \in BE(t) : M_1[(t, b)\rangle$.

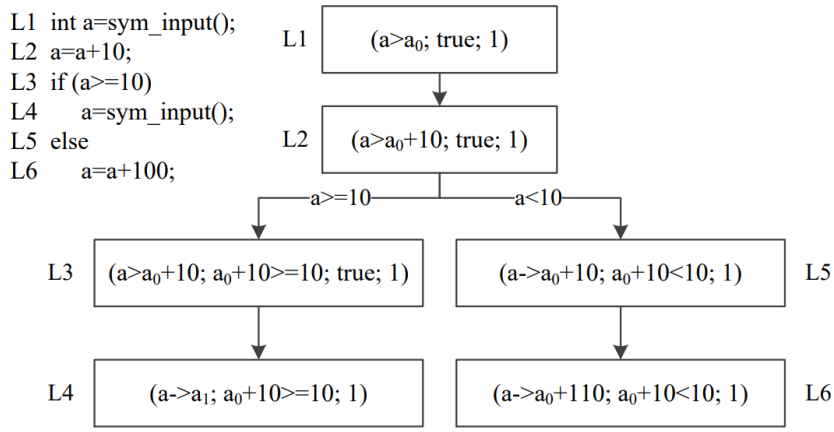


Figure 1. The update process of the parameterized variable place

4 MODEL CHECKING PDNET WITH PARAMETERIZED VARIABLES

4.1 Propositions and LTL of PDNet with Parameterized Variables

LTL describes linear temporal properties. Our approach can support the LTL-$\mathcal{X}$ formulae, so we formalize the following particular definition of propositions in PDNet with parameterized variables.

Definition 9 (Proposition of PDNet with LTL-$\mathcal{X}$ formula representation)**.** Let N be a PDNet containing parameterized variables, po a proposition, Po the set of propositions, and ψ an LTL-$\mathcal{X}$ formula, the syntax of a proposition containing parameterized variables can be defined:

$$po ::= \textit{is-fireable}(t)(t \in T) | \textit{token-value}(p_s) rop c(p_s \in P_s, c \in C(p)_{MS},$$
$$rop \in \{<, \leq, >, \geq\}).$$

Under a parameterized marking M_s, proposition semantics is defined:

$$\textit{is-fireable}(t) = \begin{cases} \textit{true}, & \text{if } \exists b : M_s[(t,b)\rangle, \\ \textit{false}, & \textit{otherwise}, \end{cases}$$

$$\textit{token-value}(p_s)\ \textit{rop}\ c = \begin{cases} \textit{true}, & \textit{if } M(p_s)\ \textit{rop}\ c\ \textit{holds}, \\ \textit{false}, & \textit{otherwise}. \end{cases}$$

LTL-$\mathcal{X}$ syntax on Po: $\psi ::= Po|\neg\psi|\psi_1 \wedge \psi_2|\psi_1 \vee \psi_2|\psi_1 \Rightarrow \psi_2|\mathcal{F}\psi|\mathcal{G}\psi|\psi_1\mathcal{U}\psi_2$ ($\neg$, $\wedge$, $\vee$ and $\Rightarrow$ are usual propositional, $\mathcal{F}$, $\mathcal{G}$, $\mathcal{U}$ are temporal operators.

For example, $\mathcal{G}$ *is-fireable*$(t) \Rightarrow \mathcal{F}$ *token-value*$(p) = 0$ implies that the number of tokens of p will be equal to 0 in some subsequent states regardless of when the transition is enabled.

4.2 Parameterized Reachability Graph for PDNet

The parameterized approach is attractive in solving the problem of parameterized variables in model checking. To enhance the expressive and analytical capabilities of PDNet, we propose a parameterized reachability graph with the following formal definitions of parameterized reachable marking and parameterized reachable marking set.

Definition 10 (Parameterized reachable marking)**.** Let $N = (\Sigma, V, P, T, F, C, G, E, I)$ be a PDNet with parameterized variables if there exists a sequence of change occurrences σ_s such that the initial parameterized marking M_{s0} can get a new parameterized marking M_s after the occurrence of σ_s, then the parameterized marking M_s is said to be reachable from the initial parameterized marking M_{s0}, i.e., $M_{s0} \xrightarrow{\sigma_s} M_s$.

Definition 11 (Parameterized reachable marking set)**.** The parameterized reachable marking set $R(M_{s0})$ of a PDNet system $N = (\Sigma, V, P, T, F, C, G, E, I)$ containing parameterized variables is a minimal set of marking satisfying the following conditions: $M(_{s0}) \in R(M_{s0})$; if $M_s \in R(M_{s0})$ and there exists $t \in T$, such that $M_s \xrightarrow{t} M'_s$, then there is $M'_s \in R(M_{s0})$.

Definition 12 (Parameterized reachability graph)**.** Let N be a PDNet with parameterized variables. The parameterized reachability graph of N is a directed graph $PRG(N) = (V, E)$, where the set of nodes of the directed graph $V = R(M_{s0})$, defining ES as the execution sequence $\langle t, \sigma, PC\rangle$, and the set of edges of the directed graph $E = \{\langle M_s, t, M'_s\rangle \cup ES | M_s, M'_s \in R(M_{s0}) \wedge M_s \xrightarrow{t} M'_s\}$; i.e., a directed graph is a graph composed of nodes identified with arcs labeled by elements in the set of variables of N.

For the parameterized reachability graph in PDNet, the process of determining whether the parameterized reachable marking is old, updating the information

stored in the parameterized reachable marking, and updating the path constraints are all different from the traditional methods of constructing reachability graphs because the parameterized reachable marking is defined. Determining whether the parameterized reachable marking is old or not by Algorithm 2. And the selection of upper bound k will be a difficult problem. Here, we use the cyclic dependency judgment algorithm [44, 45, 46] to give the upper bound k. The selection of upper bound k will significantly affect the processing efficiency of this algorithm in programs containing unbounded loops, loops, and boundary conditions of simple nested loops, which can alleviate the path explosion problem in loops to some extent. However, this loop-dependent judgment algorithm also has certain limitations: it cannot handle nonlinear loops and complex nested loops that contain dynamic boundary loops, branching conditions inside the loop, etc. The optimization of the algorithm for calculating the upper bound k will also be an important research direction for this topic in the future. The construction algorithm of the parameterized reachability graph PRG is proposed in Algorithm 2.

Algorithm 2 Construct $PRG(N)$

Use M_0 as the root node of $PRG(N)$ and label it as "*new*", with path constraint $PC = true$;

Step 1. **While** the Existence of nodes marked as "*new*" **Do**
Choose any node labeled "*new*" and set it to M;

Step 2. **If** There is a node on the directed path from M_0 to M whose marking is equal to M, For parameterized reachable marking M, reach a maximum upper bound k or terminate when identical parameterized marking exists **Then**
Change the label of M to "*old*" and return to **Step 1**;

Step 3. **If** $\forall t \in T : \neg M[t\rangle$ **Then**
Change the label of M to "*endpoint*" and return to **Step 1**;

Step 4. **For** identifies each $t \in T$ in M that satisfies $M[\rangle$ **Do**
If $L_v(t) \neq \emptyset$ **Then**;
$PC = PC \cap L_v(t)$;
4.1 According to Algorithm 1, calculate M' in $M[t\rangle M'$;
4.2 Introduce a "*new*" node in $PRG(N)$, draw a directed arc
from M to M', and label this arc with t;

Step 5. Erase the "*new*" label of node M, reset the path constraint PC to $true$, and return to **Step 1**;

4.3 Product Automaton for Parameterized Reachability Graph

For the parameterized reachability graph PRG, in the process of synthesizing the product automata, since the parameterized reachability graph nodes contain parameterized propositional states, it is not possible to solve directly whether they can be

synthesized as in the traditional product automata judgment algorithm, so here the $SAT()$ function is used to determine whether there is a feasible solution to make the parameterized propositional states synthesizable, and the constraints contained in the propositions are also added to the parameterized The constraints contained in the proposition are also added to the path constraints of the parameterized propositional state. The product automata synthesis judgment algorithm for parameterized graphs is shown in Algorithm 3, where $Label(v)$ denotes the propositional state in the parameterized reachability graph node and $L(s)$ is the set of propositions on state s in the labeled Büchi automata:

Algorithm 3 The product automaton generation algorithm for parameterized reachability graph

For Each proposition $l(s)$ in $L(s)$ **Do**
  For $Label(v)$ for each node v in the set of nodes **Do**
    If $l(s)$ contains the parameterized variable a **Then**
      Find σ and PC stored in the parameterized variable a in $Label(v)$;
      If $SAT(a.\sigma \wedge a.PC \wedge l(s) \neq false$ **Then**
        $a.PC = a.PC \wedge l(s)$;
        Synthetic product-state;
      Else Non-synthetic;
    If Proposition $l(s)$ does not contain parameterized variables **Then**
      If $lable(v) \wedge l(s) \neq false$ **Then**
        Synthetic cross-state;

To show more concretely the differences between Algorithm 2, Algorithm 3, and the traditional algorithms, we give an example of an LTL verification problem with parameterized variables in Section 4.4, which shows in detail the example graphs of the parameterized reachability graphs constructed in that case with a product automaton.

4.4 Verification Problems Based on PDNet with Parameterized Variables

Traditionally, the automata-theoretic approach for explicit model checking exhaustively explores all possible executions of the state space. The model-checking problem of LTL-$_{\mathcal{X}}$ is converted into an emptiness-checking problem [30] with the following steps:

Step 1. First model the system with parameterized variables using PDNet and construct the parameterized reachability graph $PRG(N)$ with parameterized variables;

Step 2. Describe the characteristics of the system subject to model checking using the linear temporal logic formula φ;

Step 3. Constructing Büchi automata that recognize linear temporal logic formulas φ that contain all sequences of states that violate the semantics of p;

Step 4. Constructing the parameterized reachability graph $PRG(N)$ and the product automata SP describing the Büchi automata of $\neg\varphi$, which accepts all infinite sequences of the system that are also acceptable to both the parameterized reachability graph and the Büchi automata;

Step 5. Testing whether the product automaton SP is empty, i.e., testing whether it does not accept any sequence. If SP is empty, it is proved that all runs of the system satisfy the specification p; otherwise, the system does not satisfy the specification p. Among them, Steps 4 and 5 can be handled dynamically, i.e., checking the emptiness while yielding the product automaton.

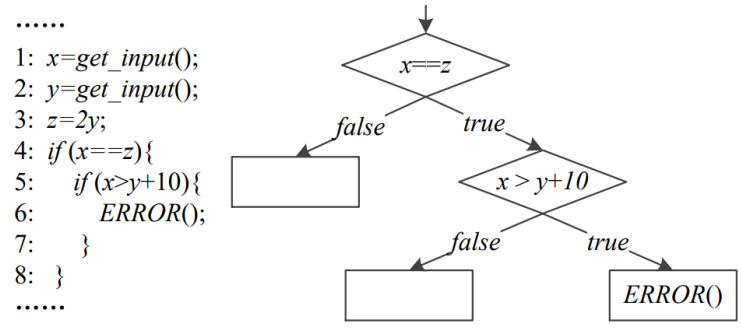
PDNet can apply an automata-theoretic approach [30], for which the marking of PDNet with parameterized variables can be generated from the initial parameterized marking and the initial state of the Büchi automaton. The acceptable paths from the initial product are extended until a product state is reached (e.g., a combined state with Büchi states). To yield the product automaton, the judgment of product automaton needs to be performed especially using Algorithm 3. Finally, all paths constitute the language accepted by the product automaton.

This example focuses on the LTL verification problems for a program containing parameterized variables. In the example program in Figure 2 a), the error location is at line 6. $ERROR()$ is an error location for safety property. Figure 2 b) represents the path branch of the example program. Here, the values of x and y are input variables by the user from the outside, and the value of z is taken concerning y. Therefore, the three variables x, y, and z are parameterized variables. The execution path of the program is shown in Figure 2 b), which is divided into three main branches, among which, if the path conditions of $x == z$ and $x > y + 10$ are satisfied at the same time, it will reach $ERROR()$. In contrast, the other two branch paths are correctly executed.

The PDNet of the example program is shown in Figure 3, with all labels on the arcs omitted for simplicity. Each transition can simulate the execution of a statement by its occurrence, and the corresponding transition occurrence can manipulate the variables represented by the place.

The state space of this PDNet is the reachability graph in Figure 4. The labeled nodes are represented by rectangles with the name of the place, and the names of the arrows on the state-labeled reachability graph correspond to the names of the transition corresponding to the occurrence of transition in the PDNet. The labels on all arcs are also omitted here for simplicity. In addition, since LTL-$\mathcal{X}$ model checking is based on infinite paths, arcs pointing to themselves are added as dashed arrows for M_3, M_5, and M_7 in Figure 4.

The LTL-$\mathcal{X}$ formula $\mathcal{G}\neg error()$ to specify the safety properties of the example program, $\mathcal{G}\neg error()$ is first converted to $is - fireable(t_3)$ in Figure 5. The node marked as $is - fireable(t_3)$ can only synchronize with the reachable marking enabled by the enabled transition t_3. The final product automaton is shown in Figure 6, and



a) Example program in C language

b) Path branch of the example program

Figure 2. Example program with parameterized variables

it can be concluded that the example program violates the security property. The occurrence sequence t_b, t'_1, t'_2, t_3 is a counterexample path in this example.

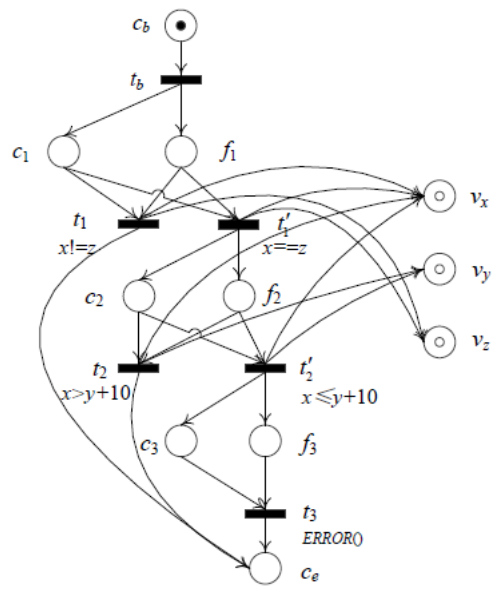


Figure 3. PDNet for the example program

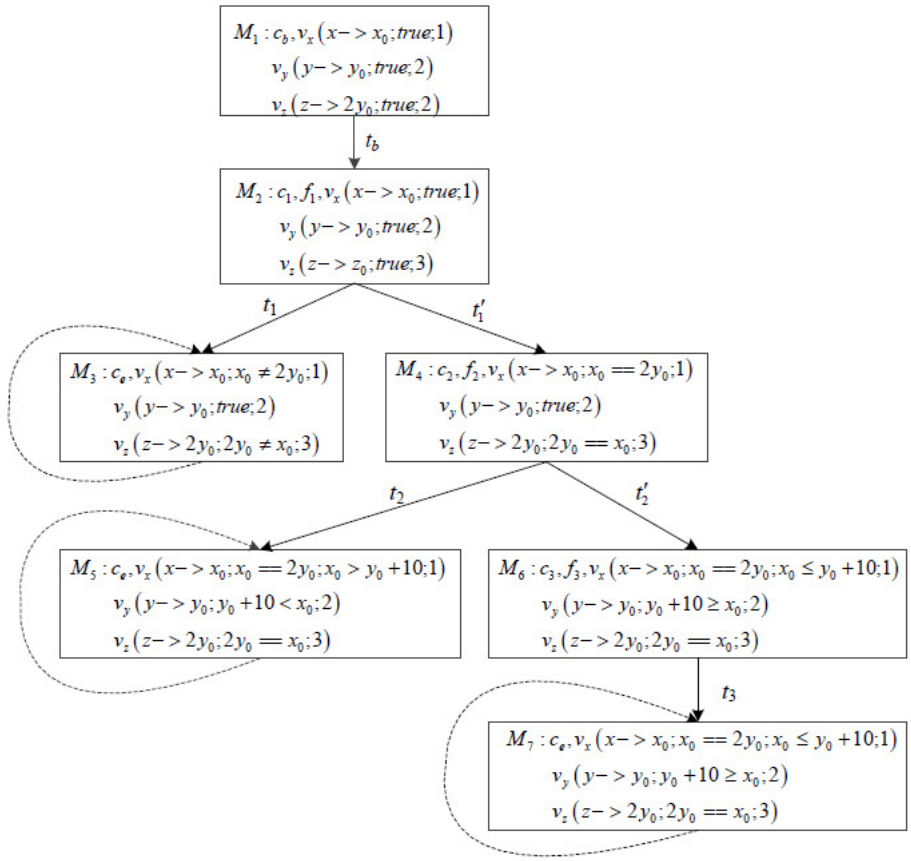


Figure 4. Parameterized state reachability graph

5 EXPERIMENTAL VERIFICATION

5.1 Experimental Benchmarks

To verify the validity of the definitions and algorithms in this paper, we construct eight typical benchmarks to evaluate the analysis capability of the system. The source code of these benchmarks includes multiple branching condition judgments on parameterized variables, repeated input judgments on parameterized variables, simple and complex computation judgments on parameterized variables, loops related to the values of parameterized variables, etc. The basic conditions are shown in Table 1 for this experiment. The benchmark is mainly judged by two aspects the tool running time and output results. In Table 1, Lines, Variables, Branches, Loops, Transitions, and Places denote the number of lines of code, variables, branches, loops, transitions, and places, respectively.

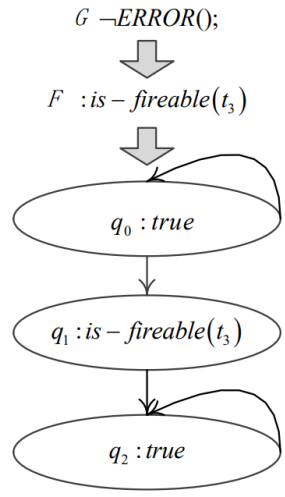


Figure 5. Büchi automaton

No.	Test program	Lines	Variables	Branches	Loops	Transitions	Places
1	Sym_Basictype	17	1	1	0	20	37
2	Sym_Branch	22	3	2	0	25	47
3	Sym_Year	21	1	1	0	23	43
4	Sym_Sum	21	2	1	0	23	44
5	Sym_Reinput	19	2	1	0	22	42
6	Sym_Loop_1	22	1	–	–	26	48
7	Sym_Loop_2	24	1	80	80	29	55
8	Sym_Loop_3	23	1	200	200	29	55

Table 1. Parameters of test program

5.2 Experimental Comparison

For each benchmark given in Table 1, the average value is taken as the experimental result after 10 runs of each benchmark algorithm because of the relatively small variation in time consumption between different runs of the same algorithm during the test. The experimental results are shown in Table 2.

Among them, the three methods used to perform comparative testing are the methods that outputs a series of test cases using the symbolic execution tool CREST and brings the benchmarks into DAMER separately for model checking, which is denoted as SymbolicExec in Table 2, the symbolic reachability graph SRG (Symbolic Reachability Graph) based model checking tool GreatSPN [47], and model checking tool CPN-AMI [48] based on Parameterized Reachability Graph PRG (Parameterized Reachability Graph).

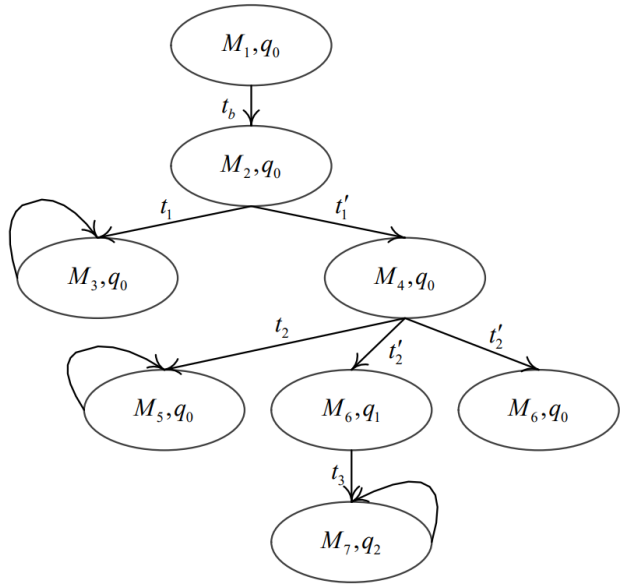


Figure 6. Product automation

t and V in the following table denote time and output results, respectively. Concretely, T and F in Table 2 denote the output result $True$ and $False$, respectively.

Test case	SymbolicExec		GreatSPN		CPN-AMI		Our method		Truth
	t	V	t	V	t	V	t	V	
Sym_Basictype	127.321	F	86.352	F	57.020	F	20.496	F	F
Sym_Branch	168.257	F	101.367	F	61.265	F	24.923	F	F
Sym_Year	126.395	F	88.215	F	58.895	F	20.586	F	F
Sym_Sum	136.257	F	93.012	F	57.958	F	21.505	F	F
Sym_Reinput	118.354	T	84.210	F	56.364	F	19.880	F	F
Sym_Loop_1	–	–	–	–	–	–	–		
Sym_Loop_2	764.258	F	397.352	T	251.035	T	104.084	T	T
Sym_Loop_3	–	–	742.362	T	422.238	F	241.715	F	F

Table 2. Comparison of experimental results

From the test results listed in Table 2, it can be seen that the SymbolicExec method misjudged or failed to judge in four test cases, including Sym_Reinput, Sym_Loop_1, Sym_Loop_2, and Sym_Loop_3; the GreatSPN method misjudged or failed to judge in two test cases, including Sym_Loop_2 and Sym_Loop_3; The CPN-AMI method does not have any misjudgment, but it also fails to judge Sym_Loop_1; the method in this paper makes correct judgments for the test cases and takes the

least time, but it also fails to judge Sym_Loop_1, which is mainly caused by the fact that the loop-dependent algorithm used in this experiment fails to judge the symbolic boundary. This is mainly caused by the fact that the loop-dependent algorithm used in this experiment cannot determine the symbolic boundary loop. It can be seen that this paper can detect programs with parameterized variables and output correct test results, which has obvious advantages in terms of test time consumption. Moreover, it can deal with branching conditions, operations, repeated input, and bounded loops of parameterized variables in programs containing parameterized variables.

For Sym_Loop_1, Sym_Loop_2, and Sym_Loop_3, all three test cases have a more serious path explosion problem, mainly caused by the loop structure present in the test cases. In Algorithm 2, the choice of the upper bound k of the loop can greatly affect the processing efficiency of this algorithm in programs containing loops. In this comparison experiment, the loop test cases are divided into the following two types according to the boundary conditions:

1. Symbolic boundary: the boundary condition expression of the loop contains parameterized variables, and the number of executions is indeterminate;
2. Constant boundary: the boundary condition expression of the loop does not contain parameterized variables, and the number of executions is constant.

Although constant-bounded loops do not execute an indeterminate number of times as symbolic-bounded loops, they also generate redundant paths leading to multiple loop expansions. In the test case, a loop dependency is implied between the parameterized variable x and the variable a such that in each loop, there are $\{x_n = x - n\}$, $\{a_n = n\}$, where, n is the number of loops. At present, we have only used a simple cyclic dependency judgment algorithm to give the cyclic upper bound k. The optimization of this algorithm will also be an important research direction for this topic in the future.

6 CONCLUSION AND FUTURE WORK

This paper improves PDNet to support parameterized variables of concurrent programs. To address the problem that it is difficult to construct the reachability graph caused by the system parameterization, we propose a new method for constructing a fully parameterized reachability graph of PDNet. We define parameterized variables on PDNet and improve the corresponding rules. The corresponding parameterized reachability graph generation algorithm is given. A PDNet-based model-checking tool that supports parameterized variables is implemented based on DAMER. The experimental results show the effectiveness of our method.

Due to parameterized variables with path information to avoid problems such as repeated execution, the amount of information in a single node of the generated reachability graph can be large. If the reachability graph is fully generated and combined with Büchi automata, the state-explosion problem is aggravated. Fu-

ture research will consider using cyclic recursive processing methods to solve this problem.

REFERENCES

[1] Mei, H.—Wang, Q. X.—Zhang, L.—Wang, J.: Software Analysis: A Road Map. Chinese Journal of Computers, Vol. 32, 2009, No. 9, pp. 1697–1710 (in Chinese).

[2] Clarke, L. A.: A System to Generate Test Data and Symbolically Execute Programs. IEEE Transactions on Software Engineering, Vol. SE-2, 1976, No. 3, pp. 215–222, doi: 10.1109/TSE.1976.233817.

[3] Weber, S.—Karger, P. A.—Paradkar, A.: A Software Flaw Taxonomy: Aiming Tools at Security. ACM SIGSOFT Software Engineering Notes, Vol. 30, 2005, No. 4, pp. 1–7, doi: 10.1145/1082983.1083209.

[4] Binkley, D.: Source Code Analysis: A Road Map. Future of Software Engineering (FOSE '07), IEEE, 2007, pp. 104–119, doi: 10.1109/FOSE.2007.27.

[5] Sekar, R.—Bendre, M.—Dhurjati, D.—Bollineni, P.: A Fast Automaton-Based Method for Detecting Anomalous Program Behaviors. Proceeding 2001 IEEE Symposium on Security and Privacy (S & P 2001), 2001, pp. 144–155, doi: 10.1109/SECPRI.2001.924295.

[6] Schumann, J. M.: Automated Theorem Proving in Software Engineering. Springer, 2001, doi: 10.1007/978-3-662-22646-9.

[7] Clarke, E. M.—Emerson, E. A.—Sifakis, J.: Model Checking: Algorithmic Verification and Debugging. Communications of the ACM, Vol. 52, 2009, No. 11, pp. 74–84, doi: 10.1145/1592761.1592781.

[8] Boulton, R. J.: Efficiency in a Fully-Expansive Theorem Prover. Technical Report. University of Cambridge, Computer Laboratory, 1994, doi: 10.48456/tr-337.

[9] Rajan, S.—Shankar, N.—Srivas, M. K.: An Integration of Model Checking with Automated Proof Checking. In: Wolper, P. (Ed.): Computer Aided Verification (CAV '95). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 939, 1995, pp. 84–97, doi: 10.1007/3-540-60045-0_42.

[10] Clarke, E. M.: Model Checking. In: Ramesh, S., Sivakumar, G. (Eds.): Foundations of Software Technology and Theoretical Computer Science (FSTTCS 1997). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 1346, 1997, pp. 54–56, doi: 10.1007/BFb0058022.

[11] Atlee, J. M.—Gannon, J.: State-Based Model Checking of Event-Driven System Requirements. IEEE Transaction on Software Engineering, Vol. 19, 1993, No. 1, pp. 24–40, doi: 10.1109/32.210305.

[12] Clarke, E. M.—Emerson, E. A.: Design and Synthesis of Synchronization Skeletons Using Branching Time Temporal Logic. In: Kozen, D. (Ed.): Logic of Programs (Logic of Programs 1981). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 131, 1982, pp. 52–71, doi: 10.1007/BFb0025774.

[13] Jensen, K.—Kristensen, L. M.—Wells, L.: Coloured Petri Nets and CPN Tools for Modeling and Validation of Concurrent Systems. International Journal on

Software Tools for Technology Transfer, Vol. 9, 2007, No. 3-4, pp. 213–254, doi: 10.1007/s10009-007-0038-x.

[14] Yang, R.—Ding, Z.—Guo, T.—Pan, M.—Jiang, C.: Model Checking of Variable Petri Nets by Using the Kripke Structure. IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol. 52, 2022, No. 12, pp. 7774–7786, doi: 10.1109/TSMC.2022.3163741.

[15] Jensen, K.—Kristensen, L. M.: Colored Petri Nets: A Graphical Language for Formal Modeling and Validation of Concurrent Systems. Communications of the ACM, Vol. 58, 2015, No. 6, pp. 61–70, doi: 10.1145/2663340.

[16] Kheldoun, A.—Barkaoui, K.—Ioualalen, M.: Formal Verification of Complex Business Processes Based on High-Level Petri Nets. Information Sciences, Vol. 385-386, 2017, pp. 39–54, doi: 10.1016/j.ins.2016.12.044.

[17] Holzmann, G. J.: The Model Checker SPIN. IEEE Transactions on Software Engineering, Vol. 23, 1997, No. 5, pp. 279–295, doi: 10.1109/32.588521.

[18] Cimatti, A.—Clarke, E.—Giunchiglia, E.—Giunchiglia, F.—Pistore, M.—Roveri, M.—Sebastiani, R.—Tacchella, A.: NuSMV 2: An Open Source Tool for Symbolic Model Checking. In: Brinksma, E., Larsen, K. G. (Eds.): Computer Aided Verification (CAV 2002). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 2404, 2002, pp. 359–364, doi: 10.1007/3-540-45657-0_29.

[19] Bolton, M. L.—Bass, E. J.—Siminiceanu, R. I.: Using Formal Verification to Evaluate Human-Automation Interaction: A Review. IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol. 43, 2013, No. 3, pp. 488–503, doi: 10.1109/TSMCA.2012.2210406.

[20] Bolton, M. L.—Bass, E. J.: Generating Erroneous Human Behavior from Strategic Knowledge in Task Models and Evaluating Its Impact on System Safety with Model Checking. IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol. 43, 2013, No. 6, pp. 1314–1327, doi: 10.1109/TSMC.2013.2256129.

[21] Katsaros, P.: A Roadmap to Electronic Payment Transaction Guarantees and a Colored Petri Net Model Checking Approach. Information and Software Technology, Vol. 51, 2009, No. 2, pp. 235–257, doi: 10.1016/j.infsof.2008.01.005.

[22] Ding, Z.—Qiu, H.—Yang, R.—Jiang, C.—Zhou, M.: Interactive-Control-Model for Human-Computer Interactive System Based on Petri Nets. IEEE Transactions on Automation Science and Engineering, Vol. 16, 2019, No. 4, pp. 1800–1813, doi: 10.1109/TASE.2019.2895507.

[23] Yin, X.—Lafortune, S.: On the Decidability and Complexity of Diagnosability for Labeled Petri Nets. IEEE Transactions on Automatic Control, Vol. 62, 2017, No. 11, pp. 5931–5938, doi: 10.1109/TAC.2017.2699278.

[24] Yang, R.—Ding, Z.—Pan, M.—Jiang, C.—Zhou, M.: Liveness Analysis of ω-Independent Petri Nets Based on New Modified Reachability Trees. IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol. 47, 2017, No. 9, pp. 2601–2612, doi: 10.1109/TSMC.2016.2524062.

[25] Ding, Z.—Yang, R.: Modeling and Analysis for Mobile Computing Systems Based on Petri Nets: A Survey. IEEE Access, Vol. 6, 2018, pp. 63038–68056, doi:

10.1109/ACCESS.2018.2878807.

[26] Jensen, K.—Kristensen, L. M.: Colored Petri Nets: A Graphical Language for Formal Modeling and Validation of Concurrent Systems. Communications of the ACM, Vol. 58, 2015, No. 6, pp. 61–70, doi: 10.1145/2663340.

[27] He, C.—Ding, Z.: More Efficient On-the-Fly Verification Methods of Colored Petri Nets. Computing and Informatics, Vol. 40, 2021, No. 1, pp. 195–215, doi: 10.31577/cai_2021_1_195.

[28] Ding, Z.—Yang, R.—Cui, P.—Zhou, M.—Jiang, C.: Variable Petri Nets for Mobility. IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol. 52, 2022, No. 8, pp. 4784–4797, doi: 10.1109/TSMC.2021.3103072.

[29] Drakaki, M.—Tzionas, P.: A Colored Petri Net-Based Modeling Method for Supply Chain Inventory Management. Simulation, Vol. 98, 2022, No. 3, pp. 257–271, doi: 10.1177/00375497211038755.

[30] Ding, Z.—Li, S.—Chen, C.—He, C.: Program Dependence Net and Its Slice for Verifying Linear Temporal Properties. CoRR, 2023, doi: 10.48550/arXiv.2301.11723.

[31] Burch, J. R.—Clarke, E. M.—McMillan, K. L.—Dill, D. L.—Hwang, L. J.: Symbolic Model Checking: 1020 States and Beyond. Information and Computation, Vol. 98, 1992, No. 2, pp. 142–170, doi: 10.1016/0890-5401(92)90017-A.

[32] Bryant, R. E.: Graph-Based Algorithms for Boolean Function Manipulation. IEEE Transactions on Computers, Vol. C-35, 1986, No. 8, pp. 677–691, doi: 10.1109/TC.1986.1676819.

[33] McMillan, K. L.: Symbolic Model Checking: An Approach to the State Explosion Problem. Ph.D. Thesis. Carnegie Mellon University, Pittsburgh, 1992.

[34] Godefroid, P.—Pirottin, D.: Refining Dependencies Improves Partial-Order Verification Methods. In: Courcoubetis, C. (Ed.): Computer Aided Verification (CAV 1993). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 697, 1993, pp. 438–449, doi: 10.1007/3-540-56922-7_36.

[35] Peled, D.: Combining Partial Order Reductions with On-the-Fly Model-Checking. Formal Methods in System Design, Vol. 8, 1996, No. 1, pp. 39–64, doi: 10.1007/BF00121262.

[36] Valmari, A.: A Stubborn Attack on State Explosion. Formal Methods in System Design, Vol. 1, 1992, No. 4, pp. 297–322, doi: 10.1007/BF00709154.

[37] Godefroid, P.: Using Partial Orders to Improve Automatic Verification Methods. In: Clarke, E. M., Kurshan, R. P. (Eds.): Computer-Aided Verification (CAV 1990). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 531, 1991, pp. 176–185, doi: 10.1007/BFb0023731.

[38] Ma, Z.—Zhu, G.—Li, Z.: Marking Estimation in Petri Nets Using Hierarchical Basis Reachability Graphs. IEEE Transactions on Automatic Control, Vol. 66, 2021, No. 2, pp. 810–817, doi: 10.1109/TAC.2020.2983088.

[39] Cousot, P.—Cousot, R.: Refining Model Checking by Abstract Interpretation. Automated Software Engineering, Vol. 6, 1999, No. 1, pp. 69–95, doi: 10.1023/A:1008649901864.

[40] Abid, C. A.—Zouari, B.: Synthesis of Controllers for Symmetric Systems. International Journal of Control, Vol. 83, 2010, No. 11, pp. 2354–2367, doi:

10.1080/00207179.2010.520415.

[41] Chiola, G.—Dutheillet, C.—Franceschinis, G.—Haddad, S.: Stochastic Well-Formed Colored Nets and Symmetric Modeling Applications. IEEE Transactions on Computers, Vol. 42, 1993, No. 11, pp. 1343–1360, doi: 10.1109/12.247838.

[42] Chiola, G.—Franceschinis, G.—Gaeta, R.: Modeling Symmetric Computer Architectures by SWNs. In: Valette, R. (Ed.): Application and Theory of Petri Nets 1994 (ICATPN 1994). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 815, 1994, pp. 139–158, doi: 10.1007/3-540-58152-9_9.

[43] Lahiri, S.—Qadeer, S.: Back to the Future: Revisiting Precise Program Verification Using SMT Solvers. ACM SIGPLAN Notices, Vol. 43, 2008, No. 1, pp. 171–182, doi: 10.1145/1328897.1328461.

[44] Tsitovich, A.—Sharygina, N.—Wintersteiger, C. M.—Kroening, D.: Loop Summarization and Termination Analysis. Vol. 6605, 2011, pp. 81–95, doi: 10.1007/978-3-642-19835-9_9.

[45] Godefroid, P.—Luchaup, D.: Automatic Partial Loop Summarization in Dynamic Test Generation. Proceedings of the 20th International Symposium on Software Testing and Analysis (ISSTA '11), 2011, pp. 23–33, doi: 10.1145/2001420.2001424.

[46] Brumley, D.—Wang, H.—Jha, S.—Song, D.: Creating Vulnerability Signatures Using Weakest Pre-Conditions. Proceedings of the 20th IEEE Computer Security Foundations Symposium (CSF '07), 2007, pp. 311–325, doi: 10.1109/CSF.2007.17.

[47] Vernier, I.: Symbolic Executions of Symmetrical Parallel Programs. Proceedings of 4th Euromicro Workshop on Parallel and Distributed Processing (PDP '96), IEEE, 1996, pp. 327–334, doi: 10.1109/EMPDP.1996.500604.

[48] Hamez, A.—Hillah, L.—Kordon, F.—Linard, A.—Paviot-Adet, E.—Renault, X.—Thierry-Mieg, Y.: New Features in CPN-AMI 3: Focusing on the Analysis of Complex Distributed Systems. Sixth International Conference on Application of Concurrency to System Design (ACSD '06), IEEE, 2006, pp. 273–275, doi: 10.1109/ACSD.2006.15.

Xiangyu Jia received her B.Sc. in computer science and technology from the Shandong University of Science and Technology, Qingdao, China, in 2021. She is currently pursuing her M.Sc. degree with the Department of Computer Science and Technology, Tongji University, Shanghai, China. Her current research interests include model checking and machine learning.

Shuo Li received her B.Sc. in software engineering from the Shandong University of Science and Technology, Qingdao, China, in 2017. She is currently pursuing her Ph.D. degree with the Department of Computer Science and Technology, Tongji University, Shanghai, China. Her current research interests include model checking, Petri nets, and formal methods.

Computing and Informatics, Vol. 42, 2023, 805–833, doi: 10.31577/cai_2023_4_805

VIGILANT SALP SWARM ALGORITHM FOR FEATURE SELECTION

N. B. Arunekumar*

Department of Artificial Intelligence and Data Science
Koneru Lakshmaiah Education Foundation, Vaddeswaram-522302, AP, India
e-mail: arunekumarbala@gmail.com

K. Suresh Joseph

Department of Computer Science, Pondicherry University Puducherry, India
e-mail: ksjoseph.csc@gmail.com

J. Viswanath

Department of Artificial Intelligence and Data Science
Madanapalle Institute of Technology and Science, AP, India
e-mail: viswaj20@gmail.com

A. Anbarasi

Department of Computing Technologies
SRM Institute of Science and Technology, Kattankulathoor, TN, India
e-mail: anbarasi.a@gmail.com

N. Padmapriya

Department of Statistics, Sri Sarada College for Women (Autonomous)
Salem, TN, India
e-mail: theen.91@gmail.com

* Corresponding author

Abstract. Feature selection (FS) averts the consideration of unwanted features which may tend the classification algorithm to classify wrongly. Choosing an optimal feature subset from the given set of features is challenging due to the complex associations present within the features. In non-convex conditions, the gradient-based algorithms suffer due to local optima or saddle points with respect to initial conditions where swarm intelligence algorithms pose a higher chance to converge over the global optima. The Salp Swarm Algorithm (SSA) proposed by Mirjalili et al. is based on the chaining behaviour of sea salps but the algorithm lacks diversity in the exploration stage. Rectifying the exploratory behaviour and testing the algorithm against the FS problem is the motivation behind this work. Three variants of the algorithm are proposed, of which the Vigilant Salp Swarm Algorithm (VSSA) inherits the vigilant mechanism in Grey Wolf Optimizer (GWO), the second variant and the third variant replace a simple crossover operator and shuffle crossover operator instead of the follower's position update mechanism used in the VSSA to form Vanilla Crossover VSSA (VCVSSA) and Shuffle Crossover VSSA (SCVSSA).

Keywords: Feature selection, optimization, k-nearest neighbors, salp swarm algorithm

Mathematics Subject Classification 2010: 68T01

1 INTRODUCTION

Feature selection is a challenging problem where two contradicting objectives of selecting the minimal number of features and achieving maximum accuracy on classification have to be attained. The feature selection discards the un-impacting or misleading features in training the algorithm for the classification. Using unwanted features for classification may deteriorate the algorithm's performance as fitting an extra dimension with respect to any learning algorithm takes considerable time due to the curse of dimensionality. The predominantly used feature selection (FS) models are of three types: filter, embedded and wrapper. The filter feature selection models are independent of the learning algorithm and rank the features based on any relationship amid the features. The ranking models are computationally low in cost. However, after ranking, choosing the n number of best features would be sub-optimal as ranking algorithms would investigate only the necessity of a single feature at a time. This phenomenon biases the feature selection only towards some specific data relationship alone. But, in a real scenario, the features may have complex dependencies. For example, a feature when being alone may not be essential but, when combined with any other features it would become a vital indicator for classification. As the complexity of the dependency between the features increases, the filter models would fail to mine out the significant feature subset. In the case of the wrapper model, the association with the classification algorithm provides feedback to the feature selection algorithm [1]. It aids the FS algorithm in procuring the

minimal and better optimal subset of features. The bio-inspired algorithms [2] can swiftly investigate the problem space and find the optimal solution in contrast to the gradient-based algorithms [3], which use only the slope of the current position. The convergence speed of the swarm intelligence algorithms is higher than the gradient following algorithms. The gradient-based algorithms may get stuck in local minima or on saddle points, paving the way to choose unwanted features and thereby deteriorating the learning algorithm's performance. The above phenomenon can be seen in the comparison table. The balanced exploration and exploitation capabilities of the swarm algorithms bestow the capability of mining the best optimal solution. A hybrid algorithm fabricated by inheriting the existing swarm algorithms' best traits will be much more efficient than their parent algorithms. The Salp Swarm Algorithm proposed by [4] uses two different mechanisms for updating a salp. One is used for updating the leader regarding the food position and the second is for updating the followers as a chain. Using a single solution for guiding may stagnate the algorithm in local optima as it lacks diversity. Introducing the influence of other eminent solutions would enhance the exploratory behaviour in the initial search and make the particles more vigilant. The GWO [5] algorithm replicates the vigilant hunting strategy of the wolves where a group of wolves surround the prey and attack them. Both the Salp Swarm Algorithm and the GWO are being used on several applications as they uncover promising solutions.

1.1 Goals

The prime goal of the proposed paper is to adopt an enhanced position update mechanism for the Salp Swarm Algorithm. The following objectives will be scrutinized to ensure enhanced performance of the proposed algorithm.

- A hybrid position update model that increases the efficiency of the guiding mechanism.
- An algorithm capable of finding a minimal and the optimal subset of features best suited for classifying the objects with high accuracy compared to the SSA.
- An algorithm that could outperform the other primarily used feature selection algorithms.
- An algorithm that aids in classifying data of different dimensions.

1.2 Organization

The paper is organized as follows: Section 1 comprises the introduction to feature selection, introduction to swarm intelligence and defines the goal. Section 2 enumerates the related work on bio-inspired and feature selection algorithms. Section 3 provides the needed preliminaries and the proposed variants. Section 4 depicts the experimentation setup. Section 5 discusses the results and analysis. Section 6 concludes the paper with the findings accomplished.

2 RELATED WORK

Feature selection is an NP-hard problem that tries to avert the curse of dimensionality. Choosing a subset of m features out of n would result in 2^n different combinations of feature subsets. Each feature in a feature vector is represented as either 1 or 0 to denote whether the respective feature is selected or not. Various algorithms have been proposed starting from the Genetic Algorithm [6], Simulated Annealing [7], Ant Colony Optimization [8] and PSO [9]. Other state-of-the-art swarm algorithms are Cuckoo Search [10], Bat Algorithm that mimics the echolocation behaviour of bats [11], Firefly Algorithm [12], Biogeography-Based Optimizer [13] and Whale Optimization Algorithm [14].

All the data observed need not necessarily be used for the classification and the data may possess several complexities such as dependency between features and irrelevant features. These feature selection algorithms pick out the essential features among the complete set and facilitate the execution of classification algorithms to provide high accuracy and low running time. The F-score [15], PCA [16], and correlation-based feature (CBF) selection [17] are some of the filter model feature selection algorithms. The filter models have no interaction with the learning algorithm and are purely dependent on the features' characteristics. The wrapper models on the other extreme works on the feedback from the learning algorithm. The elegant behaviour of the bio-inspired algorithm has attracted researchers to adopt these algorithms for wrapper feature selection algorithms. For feature selection, algorithms like hybrid genetic algorithm [18], hybrid PSO [19] with micro genetic algorithm and Gaussian mutation were used. Unlike the problems with continuous space, the binary algorithm takes values of either 0 or 1. Binary variants of the algorithms like bGWO [20], BPSO [21] and Binary Ant Lion Optimizer [22] were introduced specifically for the feature selection problems. To convert the continuous algorithms into their binary equivalent without altering any of their characteristics, the transfer functions [23] were introduced. There are totally 8 different functions that can be broadly divided into two families of S-shaped and V-shaped functions. Both the S-shaped and the V-shaped family of transfer functions map the input from the continuous range into values amid the range [0,1] which is later converted to binary value with conditions similar to Equation (11). Binary algorithms like the Binary Salp Swarm [24] and the Binary Dragonfly Algorithm [25] also used transfer functions.

3 PRELIMINARIES AND PROPOSED ALGORITHM

3.1 Brief on SSA

The SSA algorithm proposed by Ali Mirjalili et al. [4] is a population-based algorithm inspired by the swarming behaviour of the transparent jelly-like fish. This fish moves analogous to the motion exhibited by jet propulsion where the water is inhaled and exerted from its body to move forward. Along with the motion, the salps feed from

Type	Sl. No.	Dataset Name	Instances	Features		Sl. No.	Dataset Name	Instances	Features (n.o.f)
LOW DIM f < 100	1	Wine	178	13	L D	13	Movement_libras	360	90
	2	Hepatitis	155	19	HIGH DIM f ≥ 100	14	Spambase	4 601	57
	3	Vehicle	94	18		15	Arrhythmia	452	279
	4	Zoo	101	16		16	Clean1	1 593	265
	5	Heart disease	270	13		17	Hill valley	1 212	100
	6	Wisconson	682	10		18	Leukemia	72	7 070
	7	ionosphere	351	34		19	Colon	62	2 000
	8	Lung-cancer	32	56		20	Arcene	200	10 000
	9	Dermatology	366	34		21	Lymphoma	96	4 026
	10	Sonar	208	60		22	Smk_can_187	187	19 993
	11	BreastEW	569	29		23	Tox_171	171	5 748
	12	Soybean-small	47	35		24	Coil20	1 440	1 024

Table 1. Dataset description

the inhaled water by filtering out the plankton. The locomotive behaviour of the salp swarm is modelled mathematically to solve the optimization problems. They bind together as salp chains and exhibit swarming behaviour. These salps move as long chains and are attached to each other.

3.1.1 Salp Swarm Algorithm

The salps chain can be divided into two parts: the leader and the followers. The first salp is termed the leader, and the rest form the chain members or followers, as shown in Figure 2.

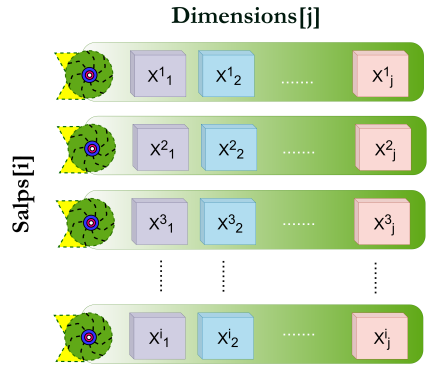


Figure 1. Representation of salp swarm in problem space

The complete population data is stored as the matrix comprising the number of individuals and the number of dimensions. All the agents and their respective features are combined to form the matrix x_j^i as in Figure 1, where i denotes the salp or agent number and j denotes the dimension. The position update of the salp members is done separately in two stages. The leader's position is updated using the Equation (2) which relies on the target food position.

$$c_1 = 2 * e^{-\left(\frac{4l}{L}\right)^2}, \tag{1}$$

$$x_j^1 = \begin{cases} T_j + c_1 * ((ub_j - lb_j) * c_2 + lb_j), & c_3 \geq 0.5, \\ T_j - c_1 * ((ub_j - lb_j) * c_2 + lb_j), & c_3 \geq 0.5, \end{cases} \tag{2}$$

$$x_j^i = \frac{1}{2}\left(x_j^i + x_j^{i-1}\right), \quad i \geq 2. \tag{3}$$

The variable x_j^1 is j^{th} dimension of the first salp. Variable T denotes the target or food, and ub and lb are the upper and lower bound respectively. The parameters c_2, and c_3 are random numbers amid $[0, 1]$ and the parameter c_1 is calculated as

Algorithm 1: SSA pseudo code

```
Initialize the population within the bound;
while current iteration ≤ total no. of iterations do
    Compute fitness of each salp;
    T = salp possessing best fitness (Target or Food);
    Update parameter c1 using Equation (1);
    for each salp particle xi do
        if i == 1 then
            Update leader salp solution using Equation (2);
        else
            Update followers on-chain using Equation (3);
        end
    end
end
Return T
```

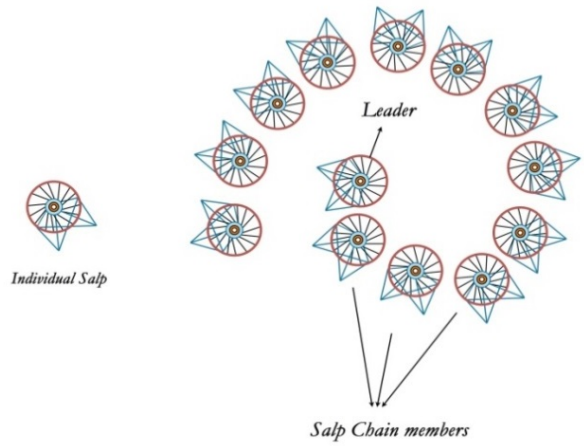


Figure 2. Salp swarm

given in Equation (1), which changes in accordance to the iteration count. The parameter c_1 is a crucial parameter that balances the algorithm between exploration and exploitation. The parameter l refers to the current iteration and L depicts the total number of iterations. The remaining salps other than the leader are being modified using Equation (3). The Pseudo code for the complete working model of the SSA is given in Algorithm 1. Several variants of the Salp Swarm Algorithm have been proposed so far. Among these variants specialized for feature selection are the bSSA [26] and iSSA [27]. In bSSA three major variants were proposed the S-shaped transfer function variants, the V-shaped transfer function variants and the

simple crossover variant. The transfer functions are one of the ways by which the values at any range are transformed into a range amid $[0, 1]$. The algorithm was run over 22 different datasets with 30 independent runs each. The iSSA algorithm used the inertia weight ω parameter from PSO and the target food. The algorithm was run over 23 different datasets with 20 independent runs each. The multi-salp chain [28] algorithm split the salps into sub-chains, updated the parameter with different strategies for each sub-chain and was run over 20 datasets with 30 iterations on each.

3.2 Proposed Algorithm

The salp chain updates the food in two phases where updating the leader position is highly critical. Based on the leader's position, the chain particles will be updated. In such cases, if the leader gets into local optima there is a huge chance for the followers to avoid promising search areas. To avert this situation the vigilant mechanism found in the GWO is introduced. Three algorithms are proposed: the first is VGSSA algorithm, the second is VCVSSA and the third is SCVGSSA.

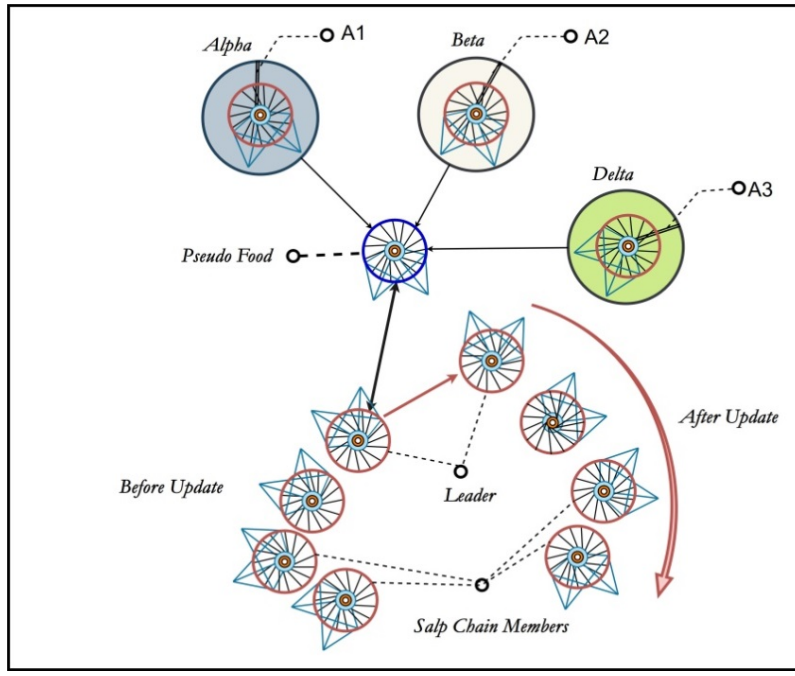


Figure 3. Vigilant SSA

3.2.1 Vigilant Salp Swarm Algorithm

Instead of relying on only one best solution as the target food, three best solutions namely Alpha, Beta and Delta are adopted. The Pseudo Food given in Equation (4) is the average of the three solutions. The Pseudo Food (PF_j) for each dimension j is replaced instead of Target Food T_j used in the SSA. The parameters r_1, r_2, r_3 are random numbers in between the range $[0, 1]$. The overall working mechanism of the VSSA is depicted in Figure 3. After the accomplishment of maximum iterations, the best solution (α) is returned as given in the pseudo-code of VSSA in Algorithm 2.

Algorithm 2: VSSA algorithm pseudo code

```
Initialize population with respect to the bounds;
while Max iterations ≥ current iteration do
    Derive fitness for each salp;
    Update α, β, δ food sources;
    α = 1st best solution;
    β = 2nd best solution;
    δ = 3rd best solution;
    PF = Compute Pseudo Food with Equation (4);
    Update c1 w.r.t. Equation (1);
    for each salp particle xi do
        if xi is leader then
            Use Leader position update as in Equation (8);
        else
            Use followers position update as in Equation (3);
Return α;
```

$$PF_j = ((A_1 * \alpha_j) + (A_2 * \beta_j) + (A_3 * \delta_j))/3, \tag{4}$$

$$A_1 = 2 * r_1, \tag{5}$$

$$A_2 = 2 * r_2, \tag{6}$$

$$A_3 = 2 * r_3, \tag{7}$$

$$x_j^1 = \begin{cases} PF_j + c_1 * ((ub_j - lb_j) * c_2 + lb_j), & c_3 \geq 0.5, \\ PF_j - c_1 * ((ub_j - lb_j) * c_2 + lb_j), & c_3 \geq 0.5. \end{cases} \tag{8}$$

3.2.2 Vanilla Crossover Vigilant Salp Swarm Algorithm (VCVSSA)

As a binary problem, feature selection either rejects or accepts a feature. Using the proposed VSSA algorithm an enhancement in the exploration of the agents can

be achieved. However, the salp chain followers still update their positions using Equation (3) which intuitively relocates the solution amid itself and its predecessor. This phenomenon can be more efficiently modelled by using a crossover operator ψ as in Equation (9). The crossover operator is predominantly used in the inheritance phase of the genetic algorithm [6, 29, 30]. The crossover operator can extract the exact features from both of its parents. The vanilla (simple) single-point crossover as given in Figure 4 inherits half of its characteristics from parent A and the rest half from parent B which is equivalent to Equation (3).

$$x_i^{t+1} = \psi(x_i, x_{i-1}). \tag{9}$$

In Equation (9), the child feature set x_i^{t+1} is the salp i at time $t+1$ which is derived from its parents, x_i the salp itself at time t and the salp's predecessor x_{i-1}.

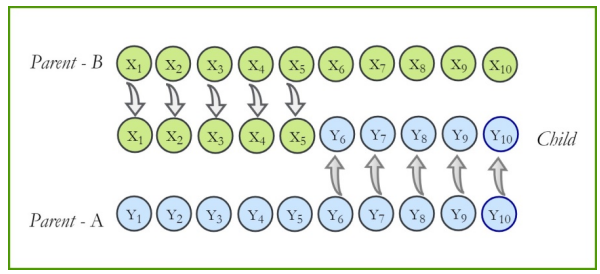


Figure 4. Simple single point crossover (vanilla crossover)

3.2.3 Shuffle Crossover Vigilant Salp Swarm Algorithm (SCVSSA)

A simple single-point crossover would be a better choice as it averts having a complete, continuous domain calculation and also depicts the behaviour of Equation (3). But, the single-point crossover has a substantial drawback of always inheriting either the left or right half of the parent as a whole.

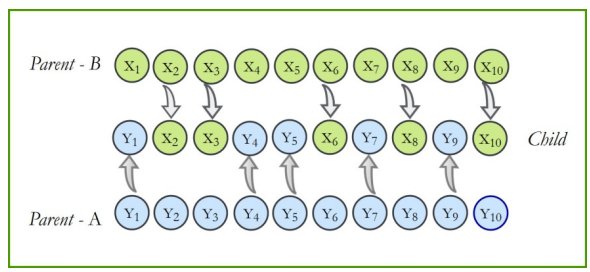


Figure 5. Shuffle crossover

The single-point crossover again introduces the paucity of inheriting multiple combinations of features. To overcome all these difficulties the proposed SCVSSA uses shuffled crossover operator φ which carries out crossover as given in Equation (10) instead of the position update by Equation (3).

$$x_j^i = \varphi(x_j^i, x_j^{i-1}). \tag{10}$$

In shuffle crossover, both the parents are shuffled with the same indices and then a single point crossover is done with the final reverse shuffling of the children to roll them back into their original indices again. The single-point crossover after the shuffling overcomes its earlier difficulties. The shuffled crossover also inputs the parents and outputs the children as done by the simple crossover. The overall mechanism of the shuffle crossover can be observed in Figure 5 and the combined flowchart for both the crossover mechanisms is given in Figure 6.

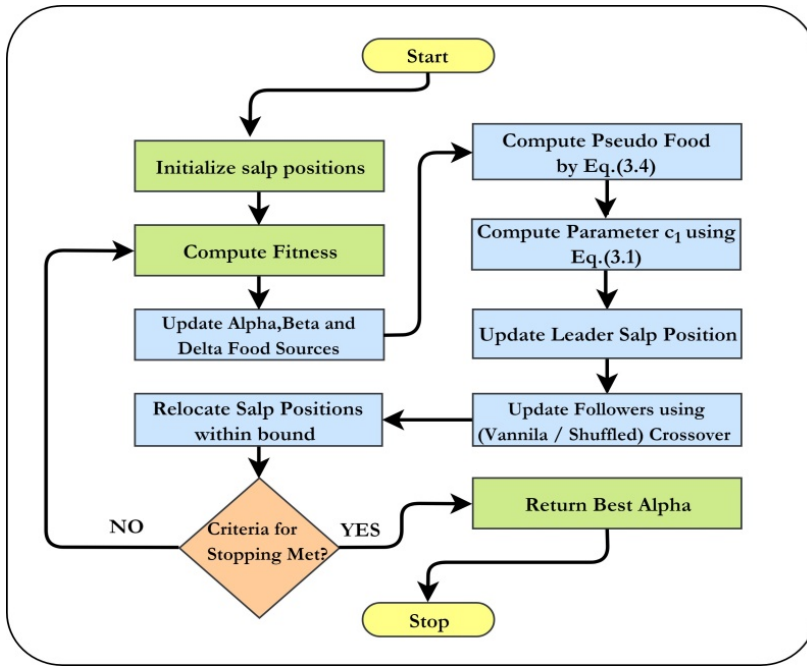


Figure 6. Flowchart for vanilla (VCVSSA)/shuffled (SCVSSA)

4 EXPERIMENTATION

4.1 Feature Representation

As discussed in Section 2, feature selection is a discrete problem. The feature sets can be modelled as binary solutions represented as an n-dimensional vector as in Figure 7 which uses 0 to reject and 1 to accept the respective feature.

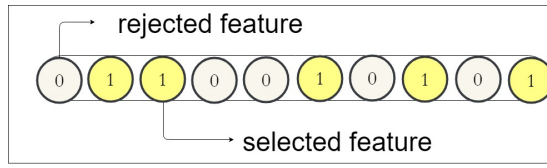


Figure 7. Feature vector representation

The algorithm is initialized and later updated over continuous domain values. To directly convert j^{th} dimension of the i^{th} continuous agent x_j^i into its respective binary agent (bx_j^i) Equation (11) is used. The converted bx_j^i is utilized to calculate the fitness of the corresponding continuous solution.

$$bx_j^i = \begin{cases} 1, & \text{if } x_j^i \geq 0.5, \\ 0, & \text{if } x_j^i < 0.5. \end{cases} \tag{11}$$

4.2 Classification Algorithm and Parameter Setup

The KNN classification used in this experiment uses certain distance measures to classify the data. A generic KNN model uses Euclidean distance as given in Equation (12) for the classification.

$$D(X_1, X_2) = \sqrt{(X_1 - X_2)^2}. \tag{12}$$

Various parameters used for the other algorithms are enumerated in the table, for the sake of fair comparison all the algorithms were implemented in the same language and compared with the same sample on each iteration.

4.3 Fitness Function

The feature selection's fitness function comprises two objectives contradicting each other. The first objective is selecting the feature that yields high accuracy for the classification algorithm and the second is selecting the least number of features. Aggregating both these objectives the fitness function which is utilized in most of the literature is being adopted as given in Equation (13). The error rate γ_R of the solution is given as (1 − accuracy). R is the number of features selected in the

solution and N is the total number of features. The hyperparameters (ρ, σ) decide the weights for the error rate and the features selected. The hyperparameter σ is given by $\sigma = (1 - \rho)$ and the other hyperparameter ρ is given as 0.99 because the reduction of error rate has to be weighted higher than the number of features. These hyperparameter values are adopted from the literature [22, 31].

$$\text{Fit} = \rho * \gamma_R(D) + \sigma * \frac{|R|}{|N|}. \tag{13}$$

4.4 Datasets

The proposed variants are tested over 24 datasets of various modalities and characteristics to prove the durability of the algorithm in versatile conditions. All the datasets have been downloaded from the standard UCI machine learning dataset repository [32] and ASU feature selection repository [33]. Very few algorithms in the literature have tested the feature selection with such enormously high-dimension datasets. Failing to test on such large dimension datasets will fail to portray the exact capabilities of the exploration and the exploitation of the algorithm. In this work, datasets of both large and small dimensions have been tested to generalize the algorithm's capability under various conditions. The datasets with feature sizes greater than 100 are termed large-dimension datasets. The number of instances in the datasets also varies in accordance with the change in dimensions. Some of the datasets possess missing values too. The above-given traits provide a challenging task of testing the exploration and exploitation ability of the algorithm such that the algorithms have to elect the optimal feature subset.

4.5 Experiment Setup

Including the raw VSSA, two other variants incorporating crossover have been proposed. All the proposed three variants are compared with the baseline Salp Swarm [4], its recently proposed hybrids bSSA [24], iSSA [27] and predominantly used feature selection algorithms bGWO [34], GOA [35], ALO [36] and PSO [37]. The general parameter setting derived from the literature [24] is used for the experimentation as enumerated in Table 2.

The feature selection algorithms are compared over three standard metrics: fitness, accuracy and number of selected features. Each algorithm is run 30 times over a dataset and its arithmetic mean is counted for the final comparison. The dataset is split into 80-20 ratio where 80 % is utilized for training the classifier and 20 % is utilized for testing it. For each of the 30 iterations performed, a unique sample of training and testing data was subjected to all the algorithms. By providing the same sample to all the algorithms for each round, the bias of the classifier with respect to the sample could be completely averted and all the algorithms would have an equal opportunity to showcase their performance.

Sl. No.	Parameters	Value
1	Population	7
2	Neighbor count in KNN	5
3	Independent run	30
4	Fitness parameters	$\rho = 0.99$, $\sigma = 0.01$
5	GOA	$c = [\text{Min} = 0, \text{Max} = 1]$, cMin = 0.0004, cMax = 1
6	PSO	$w = 1$; $c1 = 1.5$; $c2 = 2.0$;
7	(SSA/VSSA/SCVSSA), bGWO	$c1, a = [\text{Max} = 2, \text{Min} = 0]$

Table 2. Parameter configuration

5 RESULTS AND DISCUSSION

5.1 Assessment of Results

The fitness, accuracy and the number of features selected are tabulated in Tables 3, 4 and 5. The fitness value must be minimal as the error rate and no features are considered. Each cell in the table corresponds to either the mean (avg) or the standard deviation (std) for 30 independent runs on each dataset of the respective algorithms. The proposed variants are compared with the baseline, modified SSA and the other existing algorithms. On accuracy alone, the KNN without feature selection is compared.

5.1.1 Comparison over SCVSSA

Among the three proposed variants, the SCVSSA algorithm has outperformed every other algorithm over most datasets. It has been ranked 1 on both fitness and accuracy against all the other algorithms including the other proposed variants which is evident from Table 3 and Table 4. From those tables, it is also evident that the SCVSSA has outperformed all the other algorithms on 87.5 % of the datasets in terms of fitness and accuracy. Considering individually, SCVSSA has outperformed VCVSSA, bSSA [24] and PSO [37] over 95.8 % of the datasets and the VSSA, iSSA [27], bGWO [34], GOA [35], ALO [36] and baseline SSA [4] over 100 % of the datasets on accuracy and fitness. In addition, the naïve KNN classifier was also subjected to experimentation and its accuracy is compared in Table 4. It clearly shows the need for the feature selection that has increased the accuracy by 16 %. From Table 3 inference can be made that, on all datasets, the algorithm performs better. In terms of the number of features chosen, the SCVSSA is ranked second. The parameters in the fitness Equation (13) facilitate the primary goal to acquire good accuracy and the secondary goal to elect the minimal number of features.

Therefore, the SCVSSA has balanced well in accordance with the fitness equation and has tried to converge with the global minima without getting stuck or stagnating. For example, On the Soybean-small dataset, all the algorithms have achieved the

Sl.	Algorithm	SCVSSA		VCVSSA		VSSA		iSSA		bSSA		bGWO		GOA		ALO		PSO		SSA	
	Dataset	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std
1	Wine	0.0184	0.0172	**0.0171**	0.0173	0.0216	0.0181	0.0238	0.0209	0.0202	0.0184	0.0419	0.0295	0.0905	0.0373	0.0359	0.0270	0.0301	0.0250	0.0244	0.0188
2	Hepatitis	**0.0769**	0.0441	0.0790	0.0530	0.0782	0.0407	0.0878	0.0530	0.0930	0.0512	0.1138	0.0593	0.1609	0.0609	0.1111	0.0561	0.1249	0.0674	0.0795	0.0456
3	Vehicle	**0.2235**	0.0816	0.2389	0.0748	0.2427	0.0753	0.2446	0.0817	0.2439	0.0730	0.3005	0.0867	0.3506	0.1018	0.2886	0.0805	0.2609	0.0825	0.2378	0.0893
4	Zoo	0.0171	0.0226	0.0173	0.0254	0.0155	0.0251	0.0151	0.0217	**0.0148**	0.0207	0.0307	0.0371	0.0612	0.0415	0.0310	0.0332	0.0280	0.0305	0.0143	0.0241
5	Heart disease	**0.1233**	0.0325	0.1244	0.0335	0.1260	0.0284	0.1323	0.0303	0.1347	0.0293	0.1818	0.0576	0.2224	0.0544	0.1550	0.0444	0.1693	0.0594	0.1284	0.0333
6	Wisconson	**0.0161**	0.0093	0.0179	0.0097	0.0174	0.0103	0.0162	0.0094	0.0178	0.0096	0.0207	0.0098	0.0305	0.0135	0.0215	0.0108	0.0220	0.0105	0.0182	0.0103
7	Ionosphere	**0.0581**	0.0214	0.0629	0.0267	0.0659	0.0265	0.0691	0.0234	0.1020	0.0361	0.1030	0.0394	0.1327	0.0382	0.0771	0.0319	0.0871	0.0378	0.0856	0.0353
8	Lung-cancer	**0.0581**	0.0881	0.0726	0.0962	0.0823	0.0958	0.0812	0.0800	0.1322	0.1133	0.1474	0.1173	0.3238	0.1330	0.1150	0.1014	0.0975	0.1008	0.0837	0.0961
9	Dermatology	**0.0064**	0.0055	0.0086	0.0074	0.0085	0.0080	0.0115	0.0083	0.0096	0.0075	0.0132	0.0092	0.0406	0.0180	0.0174	0.0114	0.0121	0.0097	0.0079	0.0061
10	Sonar	**0.0501**	0.0313	0.0636	0.0306	0.0626	0.0296	0.0890	0.0308	0.0979	0.0399	0.1007	0.0462	0.1650	0.0499	0.1083	0.0529	0.0847	0.0428	0.0819	0.0469
11	Breast EW	**0.1349**	0.0274	0.1424	0.0304	0.1468	0.0260	0.1543	0.0227	0.1582	0.0252	0.1581	0.0281	0.2007	0.0284	0.1641	0.0252	0.1565	0.0307	0.1505	0.0264
12	Soybean-small	**0.0006**	0.0002	0.0009	0.0006	0.0008	0.0004	0.0008	0.0004	0.0032	0.0003	0.0035	0.0006	0.0041	0.0007	0.0016	0.0010	0.0021	0.0006	0.0021	0.0006
13	Movement_libras	**0.1428**	0.0394	0.1537	0.0359	0.1580	0.0380	0.1647	0.0326	0.1771	0.0377	0.1734	0.0421	0.2120	0.0392	0.1800	0.0397	0.1585	0.0396	0.1611	0.0366
14	Spambase	**0.0732**	0.0072	0.0764	0.0078	0.0772	0.0065	0.0863	0.0090	0.0836	0.0083	0.0807	0.0084	0.1088	0.0116	0.0862	0.0104	0.0804	0.0100	0.0790	0.0081
15	Arrhythmia	**0.2880**	0.0437	0.2979	0.0354	0.3010	0.0450	0.3129	0.0433	0.3241	0.0447	0.3189	0.0452	0.3640	0.0427	0.3055	0.0713	0.2986	0.0475	0.3017	0.0440
16	Clean1	0.0483	0.0224	0.0586	0.0195	0.0535	0.0198	0.0793	0.0206	0.0777	0.0226	0.0685	0.0238	0.1166	0.0214	0.0705	0.0250	**0.0431**	0.0245	0.0585	0.0234
17	Hill valley	**0.3551**	0.0219	0.3626	0.0205	0.3655	0.0266	0.3726	0.0209	0.3861	0.0198	0.3794	0.0239	0.4104	0.0235	0.3760	0.0213	0.3632	0.0214	0.3769	0.0205
18	Leukemia	**0.0133**	0.0269	0.0244	0.0406	0.0157	0.0285	0.0200	0.0353	0.0668	0.0573	0.0734	0.0658	0.1039	0.0709	0.0201	0.0464	0.0816	0.0577	0.0729	0.0636
19	Colon	**0.0332**	0.0384	0.0588	0.0554	0.0639	0.0568	0.0636	0.0493	0.1322	0.0730	0.1298	0.0836	0.1801	0.0941	0.0562	0.0598	0.1286	0.0707	0.1290	0.0677
20	Arcene	**0.0617**	0.0378	0.0832	0.0370	0.0825	0.0370	0.0764	0.0336	0.1193	0.0458	0.1243	0.0410	0.1427	0.0494	0.0749	0.0298	0.1177	0.0463	0.1170	0.0415
21	Lymphoma	**0.0399**	0.0399	0.0417	0.0393	0.0434	0.0408	0.0421	0.0394	0.0580	0.0486	0.0616	0.0518	0.0697	0.0541	0.0519	0.0421	0.0612	0.0524	0.0577	0.0519
22	Smk_can_187	**0.2107**	0.0516	0.2143	0.0564	0.2179	0.0567	0.2313	0.0461	0.2896	0.0565	0.2863	0.0591	0.3272	0.0586	0.2137	0.0689	0.2835	0.0667	0.2845	0.0600
23	Tox_171	**0.1102**	0.0462	0.1420	0.0566	0.1263	0.0610	0.1469	0.0409	0.1847	0.0527	0.1753	0.0665	0.2690	0.0674	0.1673	0.0551	0.1650	0.0577	0.1558	0.0609
24	Coil20	**0.0072**	0.0051	0.0104	0.0064	0.0098	0.0063	0.0109	0.0077	0.0189	0.0106	0.0202	0.0112	0.0249	0.0115	0.0104	0.0072	0.0151	0.0084	0.0157	0.0077
	Avg Rank	**1.5000**		3.0000		3.4375		4.9688		7.1250		10.0000		6.0625		5.9063		5.9063		5.2500	
	Final Rank	**1.0000**		2.0000		3.0000		4.0000		8.0000		9.0000		10.0000		7.0000		6.0000		5.0000	

Table 3. Fitness – comparison over proposed methods vs. existing methods

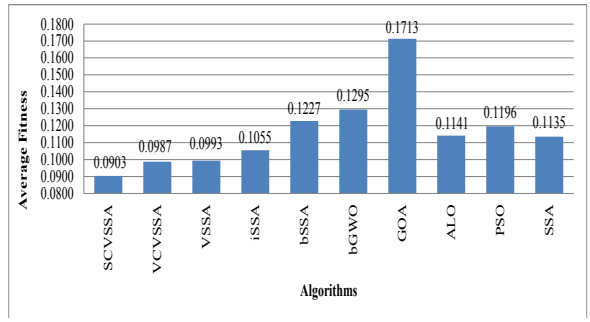


Figure 8. Comparison over average fitness

classification accuracy of 100 % and in such cases, the iSSA has failed to provide the least number of features instead, SCVSSA has chosen the least number of features. Even though SCVSSA has not been ranked one in terms of features selected, it is evident that the algorithm has provided a good reduction rate of features that provide the best accuracy which is the vital component.

5.1.2 Comparison over VCVSSA

The VCVSSA uses simple crossover despite the shuffled. This variant ranked 2, surpasses the other algorithms other than SCVSSA. The algorithm has an increased average accuracy of 15 % than the raw KNN algorithm with no feature selection. The algorithm has surpassed VSSA in over 62 % of the datasets, iSSA, ALO and SSA in over 83 % of the datasets, bSSA in over 91 % of the datasets, bGWO, GOA and KNN in over 100 % of the datasets in terms of accuracy. Sufficiently with the ordinary crossover, the algorithm could perform considerably well. However, the Shuffled crossover was proposed to improve the accuracy to some greater extent. Regarding the number of features selected, it has also managed to a good extent and is placed next to the VCVSSA. The overall performance of this variant can also be termed good when compared to the other algorithms than the VCVSSA.

5.1.3 Comparison over VSSA

The VSSA is the naïve model which did not use any crossover operator. Instead, it has used Equation (11) for acquiring the binary equivalent of a feature vector. The VSSA has the least proficiency among the proposed algorithms. But it is better than every other existing algorithm compared in Tables 3, 4 and 5.

The VSSA algorithm has outstepped the existing algorithms on 11 datasets in terms of accuracy. The algorithm has a higher accuracy for 20 of the datasets, i.e. 83 % over PSO and SSA. Likewise, higher accuracy over 87 % of the datasets has been achieved on iSSA and ALO, 91 % over bSSA and 100 % over bGWO, KNN and

Sl.	Algorithm	SCVSSA		VCVSSA		VSSA		iSSA		bSSA		bGWO		KNN		GOA		ALO		PSO		SSA	
	Dataset	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std
No	**Wine**	0.9843	0.0174	**0.9861**	0.0175	0.9815	0.0184	0.9787	0.0215	0.9833	0.0187	0.9630	0.0295	0.7204	0.0606	0.9130	0.0377	0.9685	0.0270	0.9731	0.0258	0.9787	0.0189
2	**Hepatitis**	**0.9247**	0.0451	0.9226	0.0540	0.9237	0.0411	0.9129	0.0537	0.9097	0.0518	0.8892	0.0603	0.7430	0.0750	0.8419	0.0613	0.8903	0.0572	0.8763	0.0683	0.9226	0.0461
3	**Vehicle**	**0.7772**	0.0826	0.7614	0.0754	0.7579	0.0765	0.7558	0.0827	0.7579	0.0739	0.7013	0.0876	0.5354	0.1229	0.6503	0.1030	0.7118	0.0813	0.7403	0.0838	0.7632	0.0904
4	**Zoo**	0.9856	0.0223	0.9856	0.0256	0.9873	0.0248	0.9873	0.0214	**0.9889**	0.0205	0.9738	0.0376	0.8965	0.0640	0.9431	0.0421	0.9723	0.0330	0.9755	0.0308	0.9888	0.0241
5	**Heart disease**	**0.8790**	0.0332	0.8778	0.0339	0.8765	0.0289	0.8698	0.0306	0.8679	0.0299	0.8216	0.0583	0.6667	0.0416	0.7796	0.0549	0.8469	0.0451	0.8327	0.0602	0.8741	0.0338
6	**Wisconson**	**0.9881**	0.0093	0.9864	0.0097	0.9869	0.0102	0.9878	0.0095	0.9869	0.0098	0.9854	0.0098	0.6141	0.0413	0.9742	0.0141	0.9839	0.0106	0.9822	0.0108	0.9861	0.0104
7	**Ionosphere**	**0.9427**	0.0215	0.9380	0.0271	0.9352	0.0268	0.9315	0.0236	0.9014	0.0364	0.9014	0.0398	0.8380	0.0396	0.8704	0.0385	0.9239	0.0315	0.9155	0.0381	0.9169	0.0357
8	**Lung-cancer**	**0.9429**	0.0888	0.9286	0.0975	0.9190	0.0970	0.9190	0.0812	0.8714	0.1147	0.8563	0.1187	0.4971	0.1550	0.6775	0.1346	0.8857	0.1021	0.9048	0.1016	0.9190	0.0970
9	**Dermatology**	**0.9973**	0.0055	0.9955	0.0074	0.9955	0.0082	0.9923	0.0085	0.9955	0.0074	0.9923	0.0092	0.8599	0.0364	0.9644	0.0183	0.9878	0.0114	0.9919	0.0098	0.9964	0.0061
10	**Sonar**	**0.9524**	0.0319	0.9389	0.0311	0.9405	0.0298	0.9127	0.0315	0.9063	0.0405	0.9040	0.0466	0.7667	0.0645	0.8381	0.0506	0.8944	0.0537	0.9183	0.0432	0.9214	0.0473
11	**Breast EW**	**0.8678**	0.0276	0.8605	0.0314	0.8564	0.0260	0.8476	0.0231	0.8459	0.0257	0.8468	0.0283	0.7827	0.0293	0.8023	0.0285	0.8395	0.0249	0.8468	0.0311	0.8526	0.0266
12	**Soybean-small**	**1.0000**	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	0.9433	0.0626	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
13	**Movement_libras**	**0.8588**	0.0398	0.8486	0.0369	0.8435	0.0386	0.8361	0.0333	0.8264	0.0380	0.8310	0.0425	0.7639	0.0438	0.7907	0.0396	0.8222	0.0397	0.8440	0.0399	0.8417	0.0371
14	**Spambase**	**0.9322**	0.0073	0.9290	0.0080	0.9282	0.0070	0.9170	0.0092	0.9214	0.0082	0.9261	0.0085	0.8003	0.0131	0.8952	0.0118	0.9193	0.0104	0.9239	0.0102	0.9252	0.0082
15	**Arrhythmia**	**0.7126**	0.0446	0.7025	0.0360	0.6997	0.0456	0.6864	0.0436	0.6781	0.0452	0.6846	0.0457	0.5984	0.0527	0.6373	0.0432	0.6941	0.0720	0.7026	0.0480	0.7000	0.0445
16	**Clean1**	0.9549	0.0224	0.9455	0.0195	0.9503	0.0200	0.9229	0.0206	0.9271	0.0229	0.9375	0.0240	0.8715	0.0291	0.8872	0.0216	0.9340	0.0250	**0.9608**	0.0247	0.9455	0.0236
17	**Hill valley**	**0.6443**	0.0218	0.6370	0.0208	0.6343	0.0265	0.6254	0.0211	0.6155	0.0201	0.6240	0.0244	0.5579	0.0214	0.5904	0.0239	0.6222	0.0216	0.6379	0.0216	0.6240	0.0206
18	**Leukemia**	**0.9867**	0.0271	0.9756	0.0410	0.9844	0.0287	0.9800	0.0357	0.9378	0.0579	0.9311	0.0666	0.8711	0.0699	0.9000	0.0717	0.9800	0.0468	0.9222	0.0583	0.9311	0.0643
19	**Colon**	**0.9667**	0.0388	0.9410	0.0560	0.9359	0.0574	0.9359	0.0498	0.8718	0.0738	0.8744	0.0845	0.7564	0.0949	0.8231	0.0951	0.9436	0.0604	0.8744	0.0714	0.8744	0.0685
20	**Arcene**	**0.9383**	0.0381	0.9167	0.0373	0.9175	0.0372	0.9233	0.0341	0.8850	0.0462	0.8800	0.0412	0.8308	0.0494	0.8608	0.0499	0.9250	0.0301	0.8858	0.0467	0.8867	0.0419
21	**Lymphoma**	**0.9598**	0.0403	0.9582	0.0396	0.9565	0.0410	0.9578	0.0397	0.9466	0.0490	0.9431	0.0522	0.9142	0.0663	0.9346	0.0546	0.9481	0.0425	0.9426	0.0529	0.9464	0.0524
22	**Smk_can_187**	**0.7877**	0.0521	0.7842	0.0572	0.7807	0.0571	0.7667	0.0467	0.7132	0.0571	0.7167	0.0597	0.6474	0.0597	0.6746	0.0592	0.7851	0.0691	0.7184	0.0674	0.7175	0.0606
23	**Tox_171**	**0.8914**	0.0465	0.8590	0.0571	0.8752	0.0612	0.8533	0.0416	0.8190	0.0532	0.8295	0.0673	0.6381	0.0792	0.7333	0.0681	0.8333	0.0557	0.8381	0.0583	0.8476	0.0616
24	**Coil20**	**0.9941**	0.0050	0.9916	0.0064	0.9917	0.0064	0.9900	0.0077	0.9863	0.0107	0.9859	0.0112	0.9697	0.0139	0.9799	0.0116	0.9910	0.0072	0.9890	0.0085	0.9888	0.0077
	Avg Rank	**1.2800**		3.0400		3.0400		4.7200		6.4000		7.2800		11.0000		9.6400		6.2800		6.1200		4.9200	
	Final Rank	**1.0000**		2.0000		2.0000		4.0000		8.0000		9.0000		11.0000		10.0000		7.0000		6.0000		5.0000	

Table 4. Accuracy – comparison over proposed methods vs. existing methods

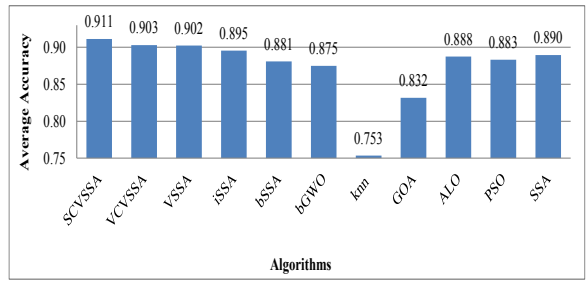


Figure 9. Comparison over average accuracy

GOA. In terms of the feature reduction, even though VSSA being ranked 4th, it has showcased a satisfactory rate of reduction in features.

5.1.4 Overall Analysis on Results with Meta-Heuristic Algorithms

Three variants were proposed, out of which one variant without crossover and the rest two with crossover operator have been proposed. All three algorithms have been compared over the three metrics of accuracy, fitness and number of features selected. Tables 3, 4 and 5 show that the variant SCVSSA that uses the shuffled crossover has gained better proficiency than the other two. Utilization of the Pseudo food has enhanced the search on exploration and exploitation stages where the leader is being re-positioned without being biased towards the best solution alone.

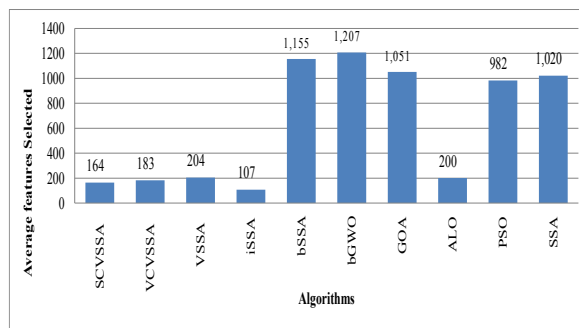


Figure 10. Comparison over the average number of features selected

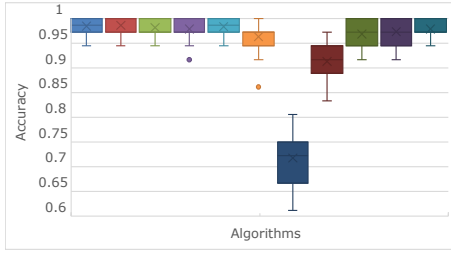
The comparison of all the datasets between the proposed algorithm and the existing algorithm is given in Figure 11 and Figure 12. The box plot depicts the median – a measure of centrality, and quartile ranges which aid the measures of dispersion, minimum and maximum values. On the algorithms such as bSSA and

Sl.	Algorithm	SCVSSA		VCVSSA		VSSA		iSSA		bSSA		bGWO		GOA		ALO		PSO		SSA	
	Dataset	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std
1	**Wine**	**3.6**	1.0	4.4	1.5	4.3	1.5	**3.6**	1.0	4.8	1.2	6.8	1.9	5.6	1.5	6.1	2.5	4.6	1.4	4.3	1.4
2	**Hepatitis**	4.6	2.6	4.4	2.8	5.1	2.6	**3.1**	1.8	6.8	2.1	7.9	1.9	8.4	2.0	4.8	3.0	4.7	1.7	5.5	2.2
3	**Vehicle**	5.3	2.2	**4.9**	1.7	5.5	2.1	5.2	1.8	7.6	1.4	8.7	1.6	7.9	1.9	5.9	2.5	6.7	1.8	6.1	1.4
4	**Zoo**	4.6	1.8	4.9	2.3	4.6	2.4	**4.1**	2.0	6.1	1.7	7.6	1.7	7.7	1.6	5.8	2.9	5.9	1.7	5.1	1.5
5	**Heart disease**	4.5	1.5	4.4	1.3	5.0	1.4	**4.3**	1.1	5.1	1.4	6.8	1.2	5.5	1.6	4.4	1.4	4.8	1.4	4.9	1.4
6	**Wisconson**	4.3	1.5	4.4	1.2	4.4	1.2	**4.1**	1.2	4.8	1.4	6.2	1.3	5.0	1.0	5.6	1.8	4.4	1.2	4.5	1.1
7	**Ionosphere**	4.8	2.2	5.3	2.9	5.9	3.2	**4.2**	2.4	15.1	2.2	18.2	3.2	14.9	2.9	6.1	5.6	11.8	3.3	11.4	2.9
8	**Lung-cancer**	8.6	6.6	10.4	7.4	11.8	7.3	**6.1**	5.1	27.3	4.0	29.1	4.6	24.9	4.0	10.2	9.5	18.1	2.3	20.1	3.4
9	**Dermatology**	**12.7**	3.5	14.0	3.1	13.8	3.0	13.2	3.3	17.5	2.6	19.1	1.9	18.3	2.8	18.1	3.8	13.8	1.9	14.6	2.4
10	**Sonar**	17.9	7.3	18.9	5.8	22.0	7.6	**15.5**	5.1	31.0	3.0	34.1	4.4	28.1	4.1	23.0	8.4	22.4	3.6	24.6	3.3
11	**Breast EW**	11.7	4.5	12.5	4.6	13.5	4.0	**9.9**	4.4	16.4	2.7	18.6	2.4	14.5	3.0	14.9	6.0	13.9	2.2	13.4	2.7
12	**Soybean-small**	**2.0**	0.6	3.1	2.1	2.7	1.5	2.7	1.5	11.3	1.0	12.1	2.1	14.3	2.4	5.7	3.7	7.2	1.9	7.3	2.1
13	**Movement_libras**	26.7	8.0	34.7	10.2	27.8	8.1	**21.9**	10.0	46.7	4.4	54.8	3.2	43.5	4.4	35.6	16.7	36.8	4.9	38.9	4.8
14	**Spambase**	34.9	5.8	34.6	7.0	34.9	6.5	**23.6**	4.9	32.9	4.1	43.3	2.7	28.8	3.4	36.3	8.3	28.8	3.4	28.5	3.5
15	**Arrhythmia**	94.5	38.2	95.6	44.2	103.2	36.3	**69.0**	30.8	150.4	8.2	185.7	11.1	137.2	9.1	76.2	51.1	116.4	9.2	129.8	7.1
16	**Clean1**	59.9	21.6	76.6	24.4	72.6	19.7	**49.7**	23.3	91.8	6.2	111.2	8.6	80.8	6.4	86.4	28.3	71.1	6.0	76.4	6.7
17	**Hill valley**	**29.5**	12.0	32.7	15.7	34.8	20.4	**17.3**	12.4	54.5	5.6	71.3	5.3	49.3	5.2	20.2	17.0	46.7	4.9	46.7	5.5
18	**Leukemia**	**99.0**	118.9	154.6	147.4	228.2	360.0	116.0	149.2	3709.8	220.9	3680.4	263.1	3494.2	35.9	190.9	223.9	3220.5	53.7	3353.8	48.5
19	**Colon**	36.5	28.4	76.1	95.0	87.1	113.5	**26.4**	34.1	1047.3	57.3	1078.1	102.7	985.3	23.0	68.6	99.1	839.4	23.2	913.2	23.4
20	**Arcene**	644.8	462.4	714.7	658.4	805.1	1080.8	**490.2**	657.7	5446.1	253.9	5507.2	473.1	4966.0	50.8	658.0	604.6	4643.1	53.5	4832.1	54.7
21	**Lymphoma**	**57.5**	49.3	106.8	105.2	137.8	178.5	104.5	98.5	2069.5	126.4	2092.0	181.9	1973.2	26.7	195.9	299.8	1766.6	28.4	1877.2	37.4
22	**Smk_can_187**	1108.9	1706.4	1339.7	1979.9	1526.6	2375.0	**499.9**	1454.4	11153.1	341.2	11562.6	1069.4	9962.1	79.8	1847.6	3017.1	9565.0	84.5	9799.0	79.1
23	**Tox_171**	1541.2	623.8	1427.1	591.0	1585.0	763.3	**971.5**	728.9	3219.3	74.8	3780.2	193.7	2855.1	32.7	1339.2	982.5	2700.6	45.4	2812.8	34.2
24	**Coil20**	135.5	64.7	206.5	119.1	154.3	87.5	**111.2**	58.6	555.3	19.5	632.3	53.4	505.3	13.6	145.9	80.0	436.5	22.0	472.2	20.1
	Avg Rank	2.4		3.6		4.4		**1.3**		8.4		9.8		8.2		5.1		5.6		6.2	
	Final Rank	2.0		3.0		4.0		**1.0**		9.0		10.0		8.0		5.0		6.0		7.0	

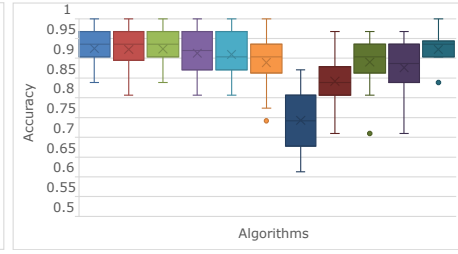
Table 5. Dimensions selected – comparison over proposed methods vs. existing methods

the bGWO the elongated whiskers and box sizes indicate a higher deviation from the median. This depicts the instability of the algorithm under various conditions and may fail to perform consistently under all conditions. From the plots, it is clearly visible that SCVSSA has less deviation and is more stable than the other algorithms in most of the datasets. The algorithm's minimum and maximum accuracies are not highly deviated from the median of the algorithm. Thus the algorithm can be termed more stable and has a good combination of exploration and exploitation under various conditions.

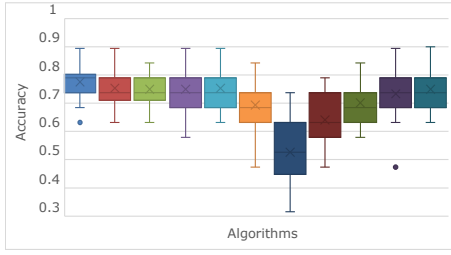
a) Color coding for the box plots



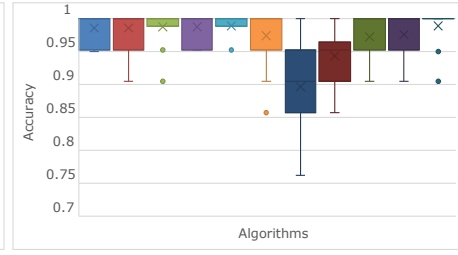
b) Wine



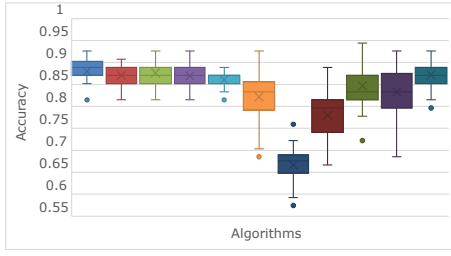
c) Hepatits



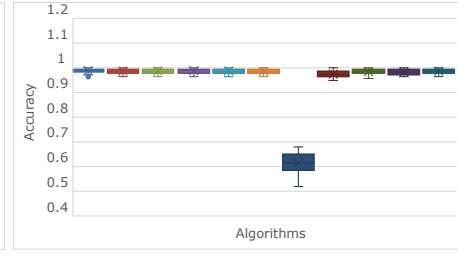
d) Vehicle



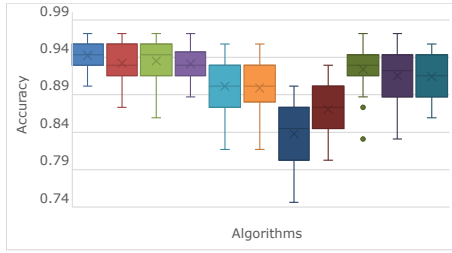
e) Zoo



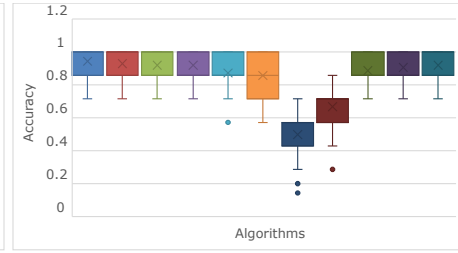
f) Heart disease



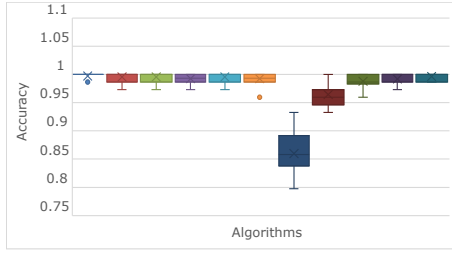
g) Wisconson



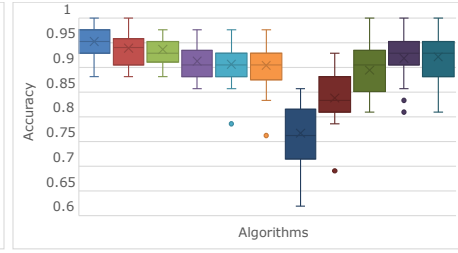
h) Ionosphere



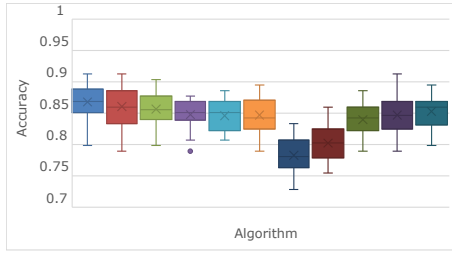
i) Lung-Cancer



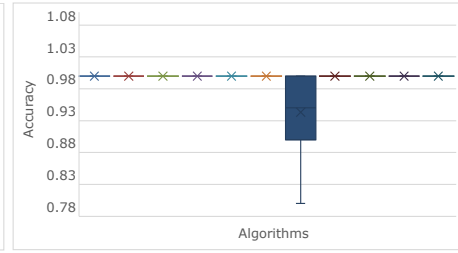
j) Dermatology



k) Sonar



l) Breast EW



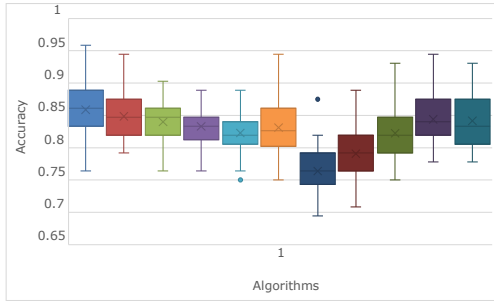
m) Soybean-small

Figure 11. Box plot comparison – proposed vs. existing meta heuristic algorithms over dataset (1–12)

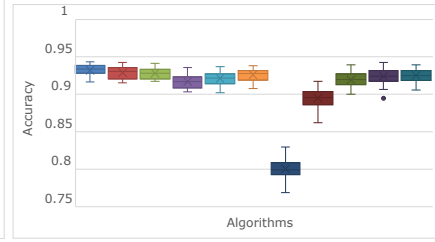
5.2 Comparison over Filter Methods

As discussed in Section 1, the filter models are independent of the classification algorithm apart from the wrapper models. These filter models are mostly used to rank the features based on its characteristics. Once the features are being ranked the first n feature would be chosen and the classification would be performed on it. The correlation-based feature selection CFS [17] uses the statistical measure of correlation to rank the feature. Other filter algorithms used for the comparison are Laplacian [38], F-score [39], relieff [40] and mutinffs [41]. The SCVSSA algorithm which is selected as the best algorithm from the previous findings is subjected to

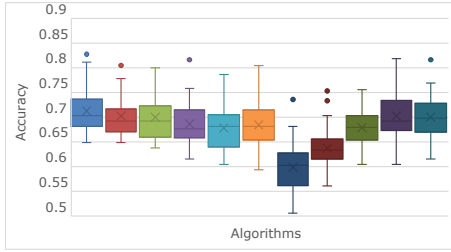
a) Color coding for the box plots



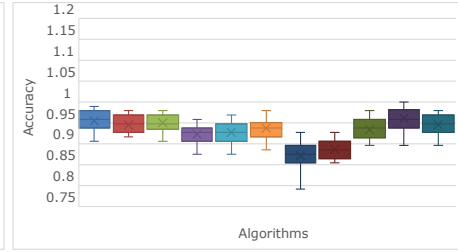
b) Movement libras



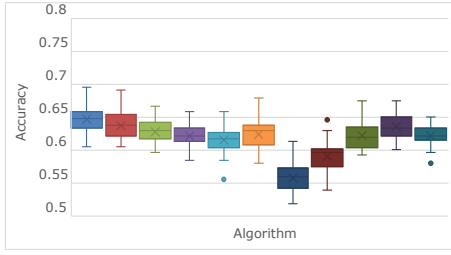
c) Spambase



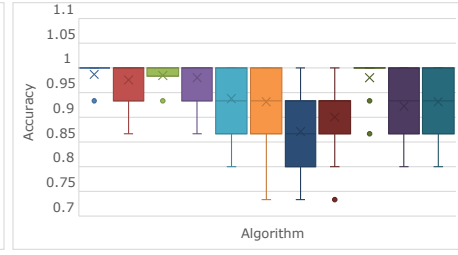
d) Arrhythmia



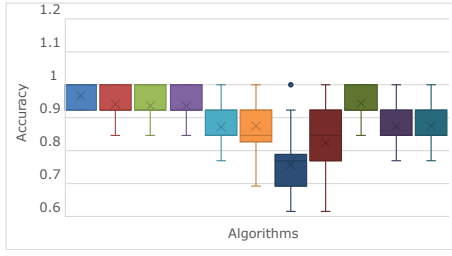
e) Clean1



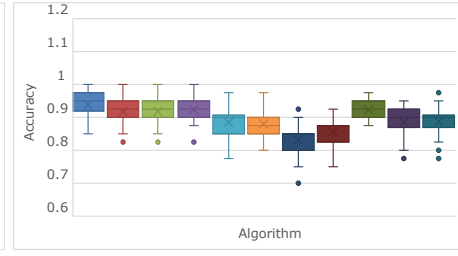
f) Hill valley



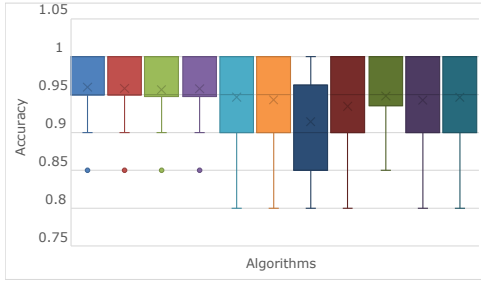
g) Leukemia



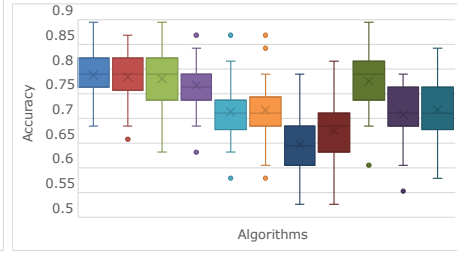
h) Colon



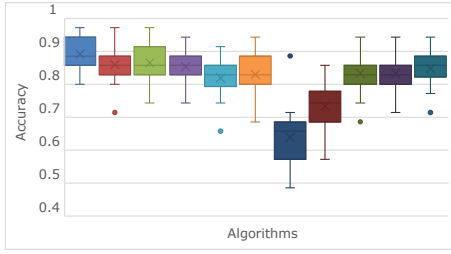
i) Arcene



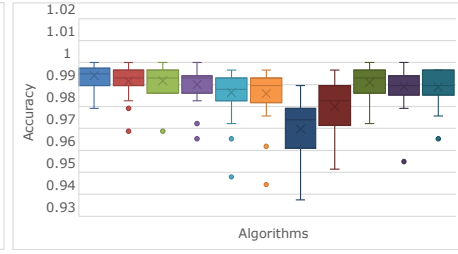
j) Lymphoma



k) Smk can 187



l) Tox 171



m) Coil20

Figure 12. Box plot comparison – proposed vs. existing meta heuristic algorithms over dataset (13-24)

test against the filter models. Once the features are ranked using the filter models the same number of n features which has been obtained by the SCVSSA has been filtered from the respective algorithms and subjected to the KNN classifier whose accuracy is enumerated in Table 6. From Table 5 it is visible that the SCVSSA has cleanly surpassed all the filter algorithms over all the datasets. From Figure 13 which summarizes the average accuracy of all the algorithms, it can be inferred that the SCVSSA is far better than the most commonly used filter models for feature selection.

6 CONCLUSION AND FUTURE WORK

In this paper, the SSA's performance has been improved by incorporating the vigilant mechanism adopted from the GWO. In addition to the above enhancement, two different crossover methods equivalent to the follower position update strategy of the SSA were applied. The main contribution of the paper is the adoption of a vigilant mechanism and shuffled crossover mechanism over the SSA. The effectiveness of this algorithm is tested by subjecting the proposed algorithms to the standard benchmark datasets downloaded from the UCI machine learning repository and ASU feature selection repository. The datasets were chosen such that they

Sl.	Algorithm	relieff		laplacian		f-Score		cfs		mutinffs		SCVSSA	
no.	Dataset	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std
1	**Wine**	0.8190	0.1054	0.7362	0.0599	0.7352	0.1169	0.5324	0.1764	0.7333	0.1116	**0.9843**	0.0174
2	**Hepatitis**	0.7387	0.0773	0.7591	0.0775	0.7785	0.0756	0.7667	0.0671	0.7903	0.0591	**0.9247**	0.0451
3	**Vehicle**	0.4778	0.0858	0.5056	0.1224	0.4722	0.0782	0.3833	0.1490	0.4926	0.1146	**0.7772**	0.0826
4	**Zoo**	0.8161	0.1022	0.8223	0.1345	0.6279	0.1451	0.4689	0.2628	0.7729	0.1008	**0.9856**	0.0223
5	**Heart disease**	0.6877	0.0736	0.6519	0.0720	0.6790	0.0666	0.6179	0.1005	0.7111	0.0434	**0.8790**	0.0332
6	**Wisconson**	0.9417	0.0221	0.6064	0.0376	0.7806	0.1731	0.7498	0.1526	0.6507	0.1121	**0.9881**	0.0093
7	**Ionosphere**	0.8376	0.0560	0.8100	0.0370	0.8895	0.0278	0.7176	0.0979	0.9010	0.0356	**0.9427**	0.0215
8	**Lung-cancer**	0.4383	0.2352	0.4195	0.1644	0.4261	0.2052	0.5144	0.1699	0.4006	0.2209	**0.9429**	0.0888
9	**Dermatology**	0.7995	0.0752	0.8365	0.1060	0.7164	0.0746	0.5973	0.1673	0.8068	0.1061	**0.9973**	0.0055
10	**Sonar**	0.7268	0.0900	0.4992	0.0643	0.7797	0.0658	0.5260	0.0768	0.7707	0.0503	**0.9524**	0.0319
11	**Breast EW**	0.7336	0.0501	0.7003	0.0413	0.7106	0.0481	0.6850	0.0753	0.7103	0.0475	**0.8678**	0.0276
12	**Soybean-small**	0.5253	0.1386	0.3577	0.1628	0.3444	0.2486	0.4098	0.2148	0.4481	0.1384	**1.0000**	0.0000
13	**Movement_libras**	0.7440	0.0538	0.7093	0.0739	0.2796	0.0830	0.2319	0.0670	0.4519	0.1069	**0.8588**	0.0398
14	**Spambase**	0.8592	0.0194	0.8809	0.0206	0.8389	0.0136	0.7816	0.0469	0.8398	0.0129	**0.9322**	0.0073
15	**Arrhythmia**	0.5884	0.0508	0.5840	0.0387	0.5617	0.0432	0.5634	0.0481	0.5939	0.0467	**0.7126**	0.0446
16	**Clean1**	0.8312	0.0448	0.8316	0.0455	0.8463	0.0287	0.7958	0.0318	0.8470	0.0288	**0.9549**	0.0224
17	**Hill valley**	0.5326	0.0306	0.5620	0.0306	0.5317	0.0279	0.5565	0.0240	0.5263	0.0333	**0.6443**	0.0218
18	**Leukemia**	0.8119	0.1036	0.8000	0.1051	0.7929	0.1033	0.8143	0.1103	0.8071	0.1030	**0.9867**	0.0271
19	**Colon**	0.6639	0.1147	0.7250	0.1279	0.7556	0.1093	0.6639	0.1443	0.6972	0.1392	**0.9667**	0.0388
20	**Arcene**	0.8092	0.0498	0.8075	0.0595	0.8017	0.0704	0.7950	0.0628	0.7925	0.0743	**0.9383**	0.0381
21	**Lymphoma**	0.8004	0.1028	0.8347	0.0801	0.7907	0.1118	0.7516	0.0895	0.8394	0.1119	**0.9598**	0.0403
22	**Smk_can_187**	0.6297	0.0758	0.6477	0.0664	0.6162	0.0907	0.0000	1.0000	0.6423	0.0776	**0.7877**	0.0521
23	**Tox_171**	0.6578	0.1060	0.6569	0.0804	0.6598	0.0744	0.6471	0.0799	0.6569	0.0652	**0.8914**	0.0465
24	**Coil20**	0.9674	0.0100	0.9306	0.0189	0.9701	0.0133	0.7925	0.0508	0.9609	0.0142	**0.9941**	0.0050
	Avg Rank	3.30		3.87		4.00		5.08		3.65		**1.00**	
	Final Rank	2		4		5		6		3		**1**	

Table 6. Accuracy – comparison over filter models

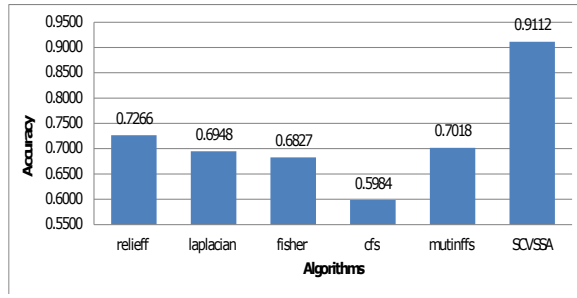


Figure 13. Comparison of average accuracy over filter models

possess various proportions of dimensions and a number of instances. To prove the proficiency of the VSSA and enhanced versions of VSSA, they were compared with the original SSA, its other hybrids and other promising meta-heuristic algorithms. The comparison and the analysis of results certainly portray that the SCVSSA could be adopted for feature selection to obtain good accuracy with the least number of features. The future direction of this work can be carried out by introducing and investigating the transfer function for the conversion of binary vectors. This wrapper model is well suited to be adopted as a pre-processing amenity for feature selection before applying a machine learning classifier.

REFERENCES

[1] Cai, J.—Luo, J.—Wang, S.—Yang, S.: Feature Selection in Machine Learning: A New Perspective. Neurocomputing, Vol. 300, 2018, pp. 70–79, doi: 10.1016/j.neucom.2017.11.077.

[2] Pedersen, M. E. H.—Chipperfield, A. J.: Simplifying Particle Swarm Optimization. Applied Soft Computing, Vol. 10, 2010, No. 2, pp. 618–628, doi: 10.1016/j.asoc.2009.08.029.

[3] Urahama, K.—Furukawa, Y.: Gradient Descent Learning of Nearest Neighbor Classifiers with Outlier Rejection. Pattern Recognition, Vol. 28, 1995, No. 5, pp. 761–768, doi: 10.1016/0031-3203(94)00142-9.

[4] Mirjalili, S.—Gandomi, A. H.—Mirjalili, S. Z.—Saremi, S.—Faris, H.—Mirjalili, S. M.: Salp Swarm Algorithm: A Bio-Inspired Optimizer for Engineering Design Problems. Advances in Engineering Software, Vol. 114, 2017, pp. 163–191, doi: 10.1016/j.advengsoft.2017.07.002.

[5] Mirjalili, S.—Mirjalili, S. M.—Lewis, A.: Grey Wolf Optimizer. Advances in Engineering Software, Vol. 69, 2014, pp. 46–61, doi: 10.1016/j.advengsoft.2013.12.007.

[6] Man, K. F.—Tang, K. S.—Kwong, S.: Genetic Algorithms: Concepts and Applications [in Engineering Design]. IEEE Transactions on Industrial Electronics, Vol. 43, 1996, No. 5, pp. 519–534, doi: 10.1109/41.538609.

[7] Mafarja, M. M.—Mirjalili, S.: Hybrid Whale Optimization Algorithm with Simulated Annealing for Feature Selection. Neurocomputing, Vol. 260, 2017, pp. 302–312, doi: 10.1016/j.neucom.2017.04.053.

[8] Kashef, S.—Nezamabadi-pour, H.: An Advanced ACO Algorithm for Feature Subset Selection. Neurocomputing, Vol. 147, 2015, pp. 271–279, doi: 10.1016/j.neucom.2014.06.067.

[9] Kennedy, J.—Eberhart, R.: Particle Swarm Optimization. Proceedings of ICNN '95 – International Conference on Neural Networks, IEEE, Vol. 4, 1995, pp. 1942–1948, doi: 10.1109/ICNN.1995.488968.

[10] Rajabioun, R.: Cuckoo Optimization Algorithm. Applied Soft Computing, Vol. 11, 2011, pp. 5508–5518, doi: 10.1016/j.asoc.2011.05.008.

[11] Arunekumar, N. B.—Kumar, A.—Joseph, K. S.: Hybrid Bat Inspired Algorithm for Multiprocessor Real-Time Scheduling Preparation. 2016 International Conference on Communication and Signal Processing (ICCSP), IEEE, 2016, pp. 2194–2198, doi: 10.1109/ICCSP.2016.7754572.

[12] Marie-Sainte, S. L.—Alalyani, N.: Firefly Algorithm Based Feature Selection for Arabic Text Classification. Journal of King Saud University – Computer and Information Sciences, Vol. 32, 2020, No. 3, pp. 320–328, doi: 10.1016/j.jksuci.2018.06.004.

[13] Simon, D.: Biogeography-Based Optimization. IEEE Transactions on Evolutionary Computation, Vol. 12, 2008, No. 6, pp. 702–713, doi: 10.1109/TEVC.2008.919004.

[14] Mirjalili, S.—Lewis, A.: The Whale Optimization Algorithm. Advances in Engineering Software, Vol. 95, 2016, pp. 51–67, doi: 10.1016/j.advengsoft.2016.01.008.

[15] Güneş, S.—Polat, K.—Yosunkaya, S.: Multi-Class F-Score Feature Selection Approach to Classification of Obstructive Sleep Apnea Syndrome. Expert Systems with Applications, Vol. 37, 2010, No. 2, pp. 998–1004, doi: 10.1016/j.eswa.2009.05.075.

[16] Wold, S.—Esbensen, K.—Geladi, P.: Principal Component Analysis. Chemometrics and Intelligent Laboratory Systems, Vol. 2, 1987, No. 1-3, pp. 37–52, doi: 10.1016/0169-7439(87)80084-9.

[17] Hall, M. A.: Correlation-Based Feature Selection for Discrete and Numeric Class Machine Learning. Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), 2000, pp. 359–366.

[18] Oh, I. S.—Lee, J. S.—Moon, B. R.: Hybrid Genetic Algorithms for Feature Selection. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 26, 2004, No. 11, pp. 1424–1437, doi: 10.1109/TPAMI.2004.105.

[19] Mistry, K.—Zhang, L.—Neoh, S. C.—Lim, C. P.—Fielding, B.: A Micro-GA Embedded PSO Feature Selection Approach to Intelligent Facial Emotion Recognition. IEEE Transactions on Cybernetics, Vol. 47, 2017, No. 6, pp. 1496–1509, doi: 10.1109/TCYB.2016.2549639.

[20] Panwar, L. K.—Reddy, S.—Verma, A.—Panigrahi, B. K.—Kumar, R.: Binary Grey Wolf Optimizer for Large Scale Unit Commitment Problem. Swarm and Evolutionary Computation, Vol. 38, 2018, pp. 251–266, doi: 10.1016/j.swevo.2017.08.002.

[21] Kennedy, J.—Eberhart, R. C.: A Discrete Binary Version of the Particle Swarm Algorithm. 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, Vol. 5, 1997, pp. 4104–4108, doi: 10.1109/ICSMC.1997.637339.

[22] Emary, E.—Zawbaa, H. M.—Hassanien, A. E.: Binary Ant Lion Approaches for Feature Selection. Neurocomputing, Vol. 213, 2016, pp. 54–65, doi: 10.1016/j.neucom.2016.03.101.

[23] Mirjalili, S.—Lewis, A.: S-Shaped Versus V-Shaped Transfer Functions for Binary Particle Swarm Optimization. Swarm and Evolutionary Computation, Vol. 9, 2013, pp. 1–14, doi: 10.1016/j.swevo.2012.09.002.

[24] Faris, H.—Mafarja, M. M.—Heidari, A. A.—Aljarah, I.—Al-Zoubi, A. M.—Mirjalili, S.—Fujita, H.: An Efficient Binary Salp Swarm Algorithm with Crossover Scheme for Feature Selection Problems. Knowledge-Based Systems, Vol. 154, 2018, pp. 43–67, doi: 10.1016/j.knosys.2018.05.009.

[25] Mafarja, M.—Aljarah, I.—Heidari, A. A.—Faris, H.—Fournier-Viger, P.—Li, X.—Mirjalili, S.: Binary Dragonfly Optimization for Feature Selection Using Time-Varying Transfer Functions. Knowledge-Based Systems, Vol. 161, 2018, pp. 185–204, doi: 10.1016/j.knosys.2018.08.003.

[26] Ren, W.—Ma, D.—Han, M.: Multivariate Time Series Predictor with Parameter Optimization and Feature Selection Based on Modified Binary Salp Swarm Algorithm. IEEE Transactions on Industrial Informatics, Vol. 19, 2023, No. 4, pp. 6150–6159, doi: 10.1109/TII.2022.3198465.

[27] Hegazy, A. E.—Makhlouf, M. A.—El-Tawel, G. S.: Improved Salp Swarm Algorithm for Feature Selection. Journal of King Saud University – Computer and Information Sciences, Vol. 32, 2020, No. 3, pp. 335–344, doi: 10.1016/j.jksuci.2018.06.003.

[28] Aljarah, I.—Mafarja, M.—Heidari, A. A.—Faris, H.—Zhang, Y.—Mirjalili, S.: Asynchronous Accelerating Multi-Leader Salp Chains for Feature Selection. Applied Soft Computing, Vol. 71, 2018, pp. 964–979, doi: 10.1016/j.asoc.2018.07.040.

[29] Faraji, R.—Naji, H. R.: An Efficient Crossover Architecture for Hardware Parallel Implementation of Genetic Algorithm. Neurocomputing, Vol. 128, 2014, pp. 316–327, doi: 10.1016/j.neucom.2013.08.035.

[30] Kaya, M.: The Effects of Two New Crossover Operators on Genetic Algorithm Performance. Applied Soft Computing, Vol. 11, 2011, No. 1, pp. 881–890, doi: 10.1016/j.asoc.2010.01.008.

[31] Arora, S.—Anand, P.: Binary Butterfly Optimization Approaches for Feature Selection. Expert Systems with Applications, Vol. 116, 2018, pp. 147–160, doi: 10.1016/j.eswa.2018.08.051.

[32] Dheeru, D.—Taniskidou, E. K.: UCI Machine Learning Repository. 2017, `http://archive.ics.uci.edu/ml`.

[33] Li, J.—Cheng, K.—Wang, S.—Morstatter, F.—Trevino, R. P.—Tang, J.—Liu, H.: Feature Selection: A Data Perspective. ACM Computing Surveys (CSUR), Vol. 50, 2017, No. 6, Art. No. 94, doi: 10.1145/3136625.

[34] Emary, E.—Zawbaa, H. M.—Hassanien, A. E.: Binary Grey Wolf Optimization

Approaches for Feature Selection. Neurocomputing, Vol. 172, 2016, pp. 371–381, doi: 10.1016/j.neucom.2015.06.083.

[35] Saremi, S.—Mirjalili, S.—Lewis, A.: Grasshopper Optimisation Algorithm: Theory and Application. Advances in Engineering Software, Vol. 105, 2017, pp. 30–47, doi: 10.1016/j.advengsoft.2017.01.004.

[36] Mirjalili, S.: The Ant Lion Optimizer. Advances in Engineering Software, Vol. 83, 2015, pp. 80–98, doi: 10.1016/j.advengsoft.2015.01.010.

[37] Eberhart, R. C.—Shi, Y.: Particle Swarm Optimization: Developments, Applications and Resources. Proceedings of the 2001 Congress on Evolutionary Computation, IEEE, Vol. 1, pp. 81–86, doi: 10.1109/CEC.2001.934374.

[38] He, X.—Cai, D.—Niyogi, P.: Laplacian Score for Feature Selection. In: Weiss, Y., Schölkopf, B., Platt, J. (Eds.): Advances in Neural Information Processing Systems 18 (NIPS 2005). MIT Press, 2005, pp. 507–514.

[39] Song, Q.—Jiang, H.—Liu, J.: Feature Selection Based on FDA and F-Score for Multi-Class Classification. Expert Systems with Applications, Vol. 81, 2017, pp. 22–27, doi: 10.1016/j.eswa.2017.02.049.

[40] Robnik-Šikonja, M.—Kononenko, I.: Theoretical and Empirical Analysis of ReliefF and RReliefF. Machine Learning, Vol. 53, 2003, No. 1-2, pp. 23–69, doi: 10.1023/A:1025667309714.

[41] Zaffalon, M.—Hutter, M.: Robust Feature Selection by Mutual Information Distributions. Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence (UCI 2002), Morgan Kaufmann Publishers Inc., 2002, pp. 577–584.

N. B. Arunekumar received his B.E. in computer science and engineering from the Anna University and M.Tech. in computer science and engineering from the Pondicherry University in 2012 and 2016. He completed his Ph.D. at the Department of Computer Science and Engineering, Pondicherry University in 2022. He is working as Assistant Professor in the Department of Artificial Intelligence and Data Science, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India. His area of interests are optimization, machine learning and deep learning.

K. Suresh Joseph received his B.E. in computer science and engineering from the Bharathiar University in 1999 and M.E. in computer science and engineering at the University of Madras in 2003. He completed his Ph.D. in computer science and engineering at the Anna University in 2013. Currently, he is working as Associate Professor in the Department of Computer Science and Engineering, Pondicherry University. His area of interest are soft computing and NDN.

J. Viswanath is currently working as Assistant Professor at the Department of Artificial Intelligence and Data Science, Madanapalle Institute of Technology and Science (MITS), Andhrapradesh. He completed his Master's degree in software engineering in 2011 at the Hindustan University, Chennai. He also completed his Bachelor's degree in information technology at SRM University, Chennai. His area of interest are software engineering, artificial intelligence and data science.

A. Anbarasi has completed her Ph.D. in computer science and engineering at the Pondicherry University. She has completed her M.Tech. in distributed computing systems at the Pondicherry Engineering College and B.E. in computer science and engineering in the V.R.S. College of Engineering and Technology. Currently, she is working as Assistant Professor in the Department of Computing Technologies, at SRM Institute of Science and Technology, Kattankulathoor, TN, India.

N. Padmapriya completed her M.Sc. (five-year integrated programme) in statistics at the Pondicherry University in 2013. She qualified CSIR-NET in the year 2017. Currently, she is working as Assistant Professor in the Department of Statistics, Sri Sarada College for Women (Autonomous), Salem-16 and pursuing her Ph.D. in the Department of Statistics, Pondicherry University.

Computing and Informatics, Vol. 42, 2023, 834–860, doi: 10.31577/cai_2023_4_834

UDP-YOLO: HIGH EFFICIENCY AND REAL-TIME PERFORMANCE OF AUTONOMOUS DRIVING TECHNOLOGY

Yonghao Liu, Hongwei Ding*, Zhijun Yang, Qianxue Xu

School of Information
YunNan University, Kunming, 650500, China
e-mail: {lyh19990202, dhw1964}@163.com

Guangen Ding

Yunnan Province Highway Networking Charge Management Co.
Kunming, 650000, China

Peng Hu

Research and Development Department, Youbei Technology Co.
Kunming, 650000, China

Abstract. In recent years, autonomous driving technology has gradually appeared in our field of vision. It senses the surrounding environment by using radar, laser, ultrasound, GPS, computer vision and other technologies, and then identifies obstacles and various signboards, and plans a suitable path to control the driving of vehicles. However, some problems occur when this technology is applied in foggy environment, such as the low probability of recognizing objects, or the fact that some objects cannot be recognized because the fog's fuzzy degree makes the planned path wrong. In view of this defect, and considering that automatic driving technology needs to respond quickly to objects when driving, this paper extends the prior defogging algorithm of dark channel, and proposes UDP-YOLO network to apply it to automatic driving technology. This paper is mainly divided into two parts:

* Corresponding author

1. Image processing: firstly, the data set is discriminated whether there is fog or not, then the fogged data set is defogged by defogging algorithm, and finally, the defogged data set is subjected to adaptive brightness enhancement; 2. Target detection: UDP-YOLO network proposed in this paper is used to detect the defogged data set. Through the observation results, it is found that the performance of the model proposed in this paper has been greatly improved while balancing the speed.

Keywords: Automatic driving technology, computer vision, object detection, image processing

1 INTRODUCTION

The target detection task is to find out the objects that people are interested in images or videos, and simultaneously detect their positions and sizes. As one of the basic problems of computer vision, target detection forms the basis of many other vision tasks, such as instance segmentation [1], image annotation [2], and target tracking [3]. From the perspective of application of detection, pedestrian detection [4], face detection [5], text detection [6], traffic light detection [7], and remote sensing target detection [8] are collectively referred to as the five major applications of target detection.

At present, the target detection algorithms are conducted in two phases: Inputting an image and generating candidate region suggestions, after classifying candidate areas and correcting coordinates, and finally detecting them. This kind of algorithm is a two-stage algorithm based on generating regional suggestions. Typical representative algorithms include R-CNN [9], Fast-RCNN [10], Faster-RCNN [11], MASK-RCNN [12], etc. The other one is the single-stage algorithm, which carries out regression analysis of neural network by directly inputting pictures, and then detecting them. This kind of algorithm regards target detection as a regression problem and does not need to generate regional suggestions. Typical representative algorithms are YOLO Series [13, 14, 15, 16, 17], SSD [18], etc. Although the accuracy of the single-stage algorithm is slightly lower than that of the former, it is favored by researchers because of its powerful real-time detection speed in such an area of pursuing real-time. Despite the above two types of algorithms have good performance in their respective fields, there are some problems in cross-domain detection. In addition, under the limited mobile devices, YOLO series algorithms cannot meet the requirements of real-time detection, so YOLOv4-tiny [19], a simplified version of YOLOv4, was born, considering both detection performance and real-time detection. YOLOv4-tiny reduces the network model and parameters based on YOLOv4, and is suitable for deployment in mobile devices with limited computing power. However, although the above algorithm has good performance when applied to other data sets, there will be some problems when applied to some specific scenes. For example, when we apply the above algorithm to the direct detection

of data sets in foggy scenes, there will be either errors in the detection category or a decline in the detection performance.

In view of this defect, this paper proposes an improved dark channel prior defogging algorithm and UDP-YOLO model, which mainly performs some image processing on the data set in foggy environment first, and then detects the defogged data set by our UDP-YOLO model. When detecting objects in fog environment, it is generally implemented by the principle of defogging firstly and then detecting. The defogging of images can be divided into two types: one is defogging by traditional learning methods, such as image enhancement or image restoration. The defogging algorithms of image enhancement include histogram equalization [20], homomorphic filtering [21], wavelet transform [22] and Retinex [23]. The defogging algorithm of image restoration includes dark channel prior the defogging algorithm. The other is to defog the image by deep learning, and the representative algorithms are DehazeNet [24], AOD-Net [25] and GCANet [22]. Although the algorithm based on deep learning is better than the algorithm based on traditional learning in image defogging, it is not suitable for unmanned driving technology because it takes a long time. Therefore, among the above-mentioned algorithms, the dark channel prior defogging algorithm based on image enhancement can achieve the defogging effect on the one hand, and has good real-time performance on the other hand. In addition, considering that the automatic driving technology will also be applied to the fog-free environment, it will waste time to defog the fog-free images, so this paper adds a fog detection algorithm based on RSV calculation [26] on the basis of this algorithm, which can first judge the quality of the images and decide whether to defog them or not. After defogging by dark channel prior algorithm, the brightness of the defogged image is dark, so the image is subjected to adaptive brightness enhancement. The above-mentioned image processing process has a general effect when it is carried out alone, and the effect is quite good when it is fused.

After image processing, the original YOLOv4-tiny model is used to detect the data set in foggy environment. Because the original model only has two prediction scales of 13×13 and 26×26, and there are a lot of small objects in our data set, it is found that the effect of the model on small object detection is not very good. Therefore, after replacing and pruning the backbone network, the neck network is also improved, multi-feature fusion is completed, and a small target detection head is added. Finally, while ensuring the performance, we added a lightweight module PPM to the model to increase the receptive field and enhance its feature extraction ability. A lightweight attention module CBAM is added to improve the performance of detection tasks. Among the measures mentioned above, our contribution can be divided into five points:

1. We extend the dark channel prior algorithm based on image enhancement, and combine it with the fog detection algorithm based on RSV calculation. Firstly, we judge the quality of the image, and then choose whether to defog all the images. In addition, the defogged image is subjected to adaptive brightness enhancement.

2. The backbone network is replaced and the number of convolution cores is reduced, and a lightweight CSP-MobileNet structure is proposed.
3. We modified the neck network, proposed a new multi-feature fusion structure, and added a small target detection head to deal with the low performance of small target detection.
4. We add PPM module to the middle area of the network to increase the receptive field of the model, so as to improve the feature extraction ability.
5. An improved CBAM attention mechanism module is added to the modified multi-feature fusion partial structure to obtain important information in the feature map.

2 RELATED WORK

2.1 Image Defogging

Image defogging is mainly divided into traditional image defogging and based on deep learning defogging. Traditional defogging algorithms include image enhancement and image restoration, and image enhancement is one of the most basic contents of digital image processing technology. In practical application, no matter what kind of device is used to collect images, the visual effect of the acquired images is not ideal due to noise, illumination, weather and other reasons. For example, the images obtained in foggy days are blurred and it is difficult to extract detailed information.

Generally, the defogging algorithm based on image enhancement does not consider the reasons of image degradation in foggy scenes, but directly processes the foggy images, so as to enhance the global characteristics or local images of the foggy images, improve the image quality, enrich the information in the images and make them look clearer. This kind of algorithm includes histogram equalization, wavelet transform, Retinex algorithm, etc. The histogram equalization algorithm makes the pixel distribution of the image more uniform and enlarges the details of the image. Wavelet transform algorithm decomposes the image and enlarges the useful part. According to the imaging principle, Retinex algorithm eliminates the influence of reflected components and achieves the effect of image enhancement and defogging. On the basis of this kind of algorithm, many improved algorithms based on the principle of image enhancement have appeared [27].

The algorithms based on image restoration are defogged by atmospheric scattering model. This kind of algorithm will first analyze the reasons that degrade the original image, and then establish a physical model to defog. The most classical algorithm is the dark channel prior algorithm. By analyzing the features of a large number of fog-free images, the prior relationship between fog-free images and some parameters in the atmospheric scattering model is found. Therefore, the detailed information of the image can be kept to a great extent to achieve the purpose of defogging, and many improved algorithms based on dark channel prior are

proposed [28, 29, 30]. Deep learning-based defogging is to train the defogging model through a large number of rich image defogging data sets as data drivers, so as to estimate the transmittance map or fog-free model for defogging. CNN or GAN can also be used to defog blurred images directly. Typical representative algorithms are Dehaze-Net and AOD-Net. Although the defogging effect of these algorithms is good, it takes a long time to process data sets.

2.2 Target Detection Based on Deep Convolution Neural Network

Early feature detection models such as Viola-Jones detector [31], HOG (Histogram of Oriented Gradients) [32] and DPM (Deformable Parts Model) [33] are constructed by integrating a series of hand-designed feature extractors. These models are characterized by a slow speed, low accuracy and poor cross-domain performance. In 2012, Krizhevsky et al. proposed AlexNet [34], an image classifier based on convolutional neural network, which achieved higher performance than the best model at that time. AlexNet used a variety of convolutional kernels to obtain image features, and also used dropout and ReLU for regularization and accelerated training respectively. Let the convolutional neural network enter the public eye, and soon caused a series of research upsurge.

Detectors based on convolutional neural networks can be divided into two categories [35]: two-stage detectors and one-stage detectors. Among them, the two-stage detector has a separate module for generating the region candidate box. The first-stage detector directly separates and locates semantic objects through intensive sampling. R-CNN is the first article in a series of two-stage detectors, which proves that CNNs can greatly improve the performance. R-CNN uses a region proposals CNN module with an unknown category to transform detection into classification and location problems. He et al. proposed to use SPP [36], a pool layer of spatial pyramid, to process pictures with arbitrary size and width ratio. This network reduces the amount of computation by shifting convolution layer and adding pooling layer, so that the network does not depend on size. Because both R-CNN and SPP-Net are trained separately in multiple stages, Faster-RCNN solves this problem by creating a single end-to-end trainable system, which is 146 times faster than R-CNN model. Lin et al. considered that in the face of small target detection, image pyramids would be used for multiple levels to obtain feature pyramids, but the calculation time would be correspondingly increased. Therefore, the feature pyramid network [37] is proposed, which adopts the top-down horizontal connection structure to construct high-level semantic features on different scales. Dai et al. proposed a method combining R-FCN [38] with Faster R-CNN to solve the translation invariance problem of convolutional neural network, and realized a fast and more accurate detector. Mask-CNN is based on Faster R-CNN, which adds a branch for parallel pixel-level target instance segmentation. DetectoRS [39] combines the above-mentioned systems to improve the performance of detectors, and is equipped with the most advanced two-stage detectors. Its RFP and SAC modules are universal and can be used in other detection models. Although the proposed two-stage

detector has good performance in target detection, it is not suitable for real-time detection because of its numerous deep convolution neural network.

Considering that the speed of two-stage detectors is really slow, the first-stage detectors directly classify and locate semantic targets through intensive sampling, and they use predefined boxes with different proportions and aspect ratios to locate targets. YOLOv1 reconstructs the detection problem, regards it as a regression problem, and directly predicts image pixels as targets and their bounding box attributes. SSD is the first one-stage detector that can maintain its performance and be compatible with real-time, but its performance for small target detection is somewhat difficult. YOLOv2 replaces the backbone network on the basis of YOLOv1, and combines a variety of technologies, such as adding BN to improve convergence, and training classification and detection systems to improve the number of detection categories. The accuracy and speed have been improved. Although the first-stage detector has achieved good results in speed, its performance is a bit low. The reason is that the background class is unbalanced. So Lin et al. proposed a modified cross entropy loss in RetinaNet [40] detector to solve this problem. YOLOv3 is based on YOLOv2 and integrates various technologies, such as data enhancement, multi-scale training, batch standardization, etc. Duan et al. proposed CenterNet [41] to model the object as a point, and the input image generates heatmap through FCN, and the peak value of heatmap corresponds to the center of the detected object. EfficientDet [42] constructed the idea of an extensible detector with higher accuracy and efficiency, and introduced effective multi-scale features, BiFPN and model scaling. YOLOv4 model combines various methods to design a target detector that can work quickly and easily in the existing system. Using the bag-of-freebies method, it only increases the training time without affecting the reasoning time.

2.3 Feature Fusion

Feature fusion is an important method in the field of pattern recognition. In many jobs, fusing features of different scales is an important means to improve detection performance. The low-level features have higher resolution and contain more location information and detail information, because of less convolution, the semantics are lower and the noise goes up. High-level features semantic information is stronger, but the resolution is low and the perception of details is poor. By fusing the features of high and low levels, we can make use of various image features, realize the complementary advantages of multiple features, and obtain more robust and accurate recognition results.

At present, the most common feature fusion methods are FPN, PANet [43] and NAS-FPN [44]. Among them, FPN uses semantic information of low-level and high-level features at the same time, and fuses features of different levels to achieve the prediction effect. Moreover, the prediction is performed separately on each fused feature layer, which is different from the conventional feature fusion method. PANet is a feature fusion structure proposed in YOLOv4, which adds a layer of FPN network from bottom to top, extracts features from each feature layer

for each proposal, and finally obtains the features to be detected by convolution-upsampling and full connection layer fusion. NAS-FPN mainly reorganizes feature maps with multiple scales, and then performs merging cell operation on them. This operation is divided into three steps: First, two candidate feature layers are selected as input feature layers, then select the resolution of the output feature, and finally, a binary operation is selected to integrate the two input feature layers into a new output feature layer. After completing this series of operations, the cyclic operation continues to obtain the final output feature layer for detection.

2.4 Attention Mechanism

With the development of science and technology, more and more information comes to us, and there is a large amount of information around us all the time. However, the information we receive in a limited time is limited, but researchers have found that the human visual system has a strong visual information processing capability in a limited field of vision. When we process information in the early stage, we will focus our attention on the important things. This choice allows us to reduce the amount of information to be processed, so that we can suppress unimportant stimuli when processing complex visual information, and provide easier and more relevant new information for higher-level perceptual reasoning and more complex visual processing tasks (such as target recognition, target classification, video comprehension, etc.). In view of this advantage, researchers put forward the idea of attention mechanism. The main idea of attention mechanism is to get the difference of the importance of each feature map through some measures, so as to use more resources for more important tasks, and use the results of tasks to guide the weight update of feature maps in reverse, thus completing the corresponding tasks efficiently and quickly.

At present, the attention mechanisms proposed are mainly divided into two types: single-channel attention and multi-channel attention. There is only one module in single-channel attention to obtain attention in the channel, and the representative networks mainly include SE-Net [45] and ECA-Net [46]. The main idea of SE-Net is to estimate the loss function value LOSS through the network model, so as to learn the feature weight. Generally speaking, the weight of the feature graph with obvious task effect becomes larger, while the weight of the feature graph with no obvious or no effect becomes smaller, and then the model is trained to achieve better results. ECA-Net is an improvement on SE-Net, and proposes a local cross-channel interaction strategy without dimensionality reduction and a method of adaptively selecting the size of one-dimensional convolution kernel. More accurate attention information is obtained by summarizing cross-channel information in one-dimensional convolution layer. There are two modules in the multi-channel attention mechanism, which mainly capture the attention between channels and feature pixels. The representative networks mainly include SK-Net [47], CBAM [48] and DA-Net [49]. SK-Net is an attention mechanism based on convolution kernel, that is, by comparing the importance of different images passing through different

convolution kernels. CBAM module puts forward that the channel of feature image not only contains a lot of attention information, but also contains rich attention information inside the channel, that is, between the pixels of feature image. Therefore, CBAM has built two modules, CAM and BAM, to collect the attention in the channel and empty space respectively, and then synthesize the collected attention to avoid wasting the attention in the space. Although the idea of DA-Net (Dual Attention Network) network is the same as that of CBAM, its way of obtaining two channels of attention information is different from that of CBAM, and it is obtained through parallel mode.

2.5 Model Pruning

In order to improve the performance of the model while maintaining the speed of the model, some pruning measures are taken to the model. Pruning method can explore the redundancy of model weights and try to trim redundant and non-critical weights [50, 51]. Model pruning is mainly divided into unstructured pruning [52] and structured pruning [53]. Unstructured pruning mainly changes the combined structure of neurons in the single layer of the network model, and its representative pruning includes fine-grained pruning [54], vector pruning [55] and nuclear pruning [56]. Although this kind of pruning can achieve a high compression rate, while maintaining a high performance. But it needs enough hardware structure to support sparse operations. Structured pruning is to change the structural characteristics of the network model, so as to achieve the effect of compressing the model. Its representative pruning includes filter pruning [57]. Although there are many types of model pruning, the main purpose is to prune the neural network structure, and the general ideas can be summarized into three types: standard pruning, pruning based on sub-model sampling, and pruning based on search. The idea of standard pruning is to carry out pre-training, then pruning, then fine-tuning, and then repeat the above processes in turn, and finally get a suitable pruning structure. Sub-model-based sampling process is to randomly sample the trained model according to the pruning target, and then prune each sampled network structure to obtain the sampling model and evaluate the best pruning model. Search-based pruning is based on unsupervised learning or semi-supervised learning algorithm, and the optimal substructures are searched by selecting pruning target.

3 MODEL DESIGN

3.1 YOLOv4-Tiny

Figure 1 shows the model structure of YOLOv4-tiny, which is simplified based on YOLOv4 model. The model consists of three parts: Backbone network to extract features, feature pyramid network to fuse features, and YOLO head to predict the acquired features.

The backbone network of this model is CSPDarknet53-tiny network, which is mainly composed of Conv and CSPBlock. Conv not only performs convolution operation on it, but also performs batch standardization and activation function operation, in which BN (Batch Norm) is used in batch standardization, and the activation function is modified to Leaky ReLU (Leaky Rectified Linear Unit). In the CSPBlock module, CSP-net is mainly used. In fact, the original residual block stack is split into two parts: the main part continues to stack the original residual block, and the other part is like a residual edge. After a small amount of processing, it is directly connected to the last two parts, and the final output is obtained by merging them. The FPN structure can fuse feature maps of different scales, which can not only ensure the rich semantic information of the deep network, but also obtain the geometric details of the low-level network, so as to strengthen the ability of feature extraction. YOLO head predicted the features of the fused feature map, and finally formed two prediction scales of 13×13 and 26×26. Although YOLOv4-tiny model has good detection performance and real-time performance on VOC and COCO data sets, its effect is not so optimistic if the model is applied to the target detection of autonomous driving technology in foggy scenes. Therefore, this paper proposes UDP-YOLO network for this defect and combines it with the defogging algorithm. Firstly, the improved defogging algorithm is used to process the image of the data set. Then, CSP-MobileNet structure is introduced into the backbone network, and a new idea of multi-feature fusion is proposed. The receptive field module is added to the extracted feature maps with dimensions of 128×128, 256×256, and 512×512. Finally, attention mechanism is added to some stages of feature fusion.

3.2 Image Processing

3.2.1 Judge Whether There is a Fog or Not

Considering that the defogging of data sets is a time-consuming process, and when the data sets contain normal pictures, defogging will make the high-frequency components contained in them change greatly, and the detection effect will be poor. Therefore, we need to improve the real-time performance and effectiveness of our improved model in foggy scenes. Firstly, the fog detection algorithm is used to calculate the ratio of saturation (S) and color value (V) of the picture, and then the defogging process is judged, as shown in Figure 2. Firstly, find out the general driving direction of vehicles, and find out their intersection point by extending the driving direction, which is the vanishing point proposed by us. Then, take this point as the center to select an area, which is called ROI box, and then calculate the ratio of saturation (S) and color value (V) in HSV color model domain in this box. In addition, considering that setting one ROI box may lead to inaccurate results, four ROI boxes are set, and the ratio of each ROI box is calculated separately and the average value is added to make a more reasonable fog judgment, which is defined as

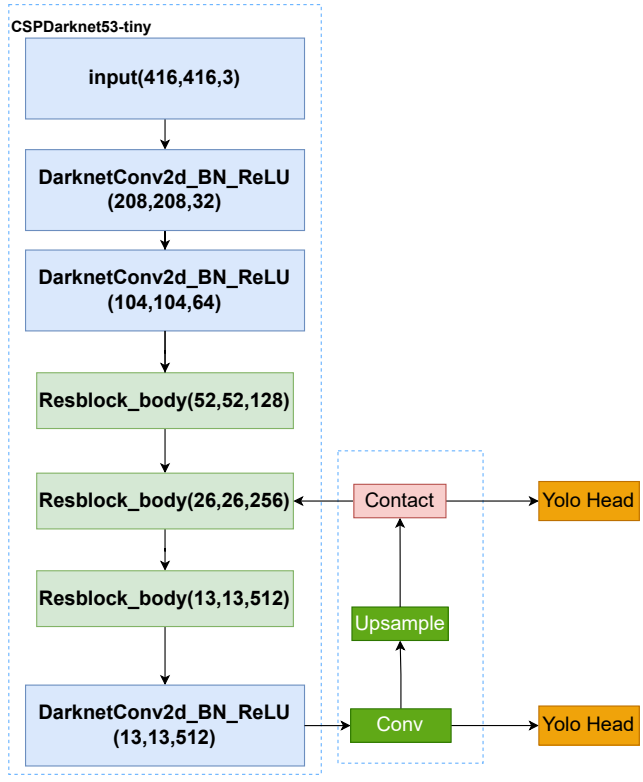


Figure 1. YOLOv4-tiny model structure

Equation (1) and Equation (2):

$$u(i) = \frac{s(i)}{v(i)}, \tag{1}$$

$$u = \frac{\sum_{i=1}^{4} u(i)}{4}, \tag{2}$$

where i is the selected i^{th} ROI box, $s(i)$ is the saturation of the i^{th} box, and $v(i)$ is the color value of the i^{th} box. $u(i)$ represents the ratio of the saturation of the i^{th} frame to the color value of the i^{th} frame, and u represents the average value of $u(i)$ of the four frames obtained respectively. It is compared by comparing the value with the set prior value $u0 = 3.5$. If the value is greater than 3.5, defogging is required, otherwise, it is not required.

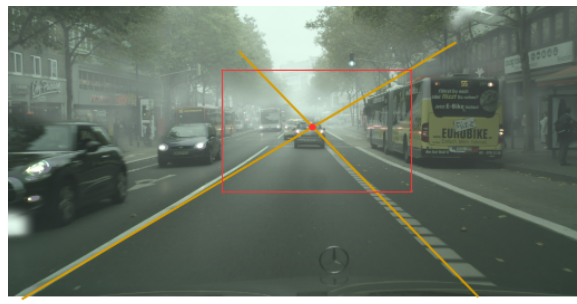


Figure 2. Judgment principle diagram of fog

3.2.2 Improve the Ability of Fog Environment Detection

After judging whether the image is foggy or not, the foggy image is restored, which uses the atmospheric scattering model to defog the image. The atmospheric scattering model is shown in Figure 3.

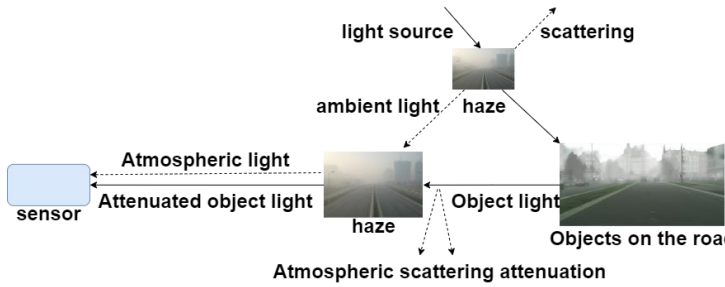


Figure 3. Atmospheric scattering model

By modeling the imaging process of fog scene, we can get:

$$F(x) = J(x)t(x) + A(1 - t(x)). \tag{3}$$

Here, x represents a certain pixel in $F(x)$, $F(x)$ represents a foggy image scattered by atmospheric particles and reaching the sensor, $J(x)$ is a clear fog-free image that is not scattered by atmospheric particles, A is atmospheric illumination, and $t(x)$ represents the transmittance in a foggy scene. In the above formula: $t(x) = e^{-\beta d(x)}$ where β represents the atmospheric scattering coefficient and $d(x)$ represents the scene depth.

Based on the atmospheric scattering model, priority of an image defogging algorithm based on dark channel is proposed. The theoretical basis of this algorithm is that when the image is taken in the normal weather, at least one channel in RGB image will have pixel values with intensity approaching 0 after the sky area of the

image is removed. Therefore, take any shot image and define it as $J(x)$, it can also be written as:

$$J^{dark}(x) = \min_{t\in\Omega(x)}(\min, J^c(t)) \quad (4)$$

In Equation (4), J^c is the image of a certain color channel in RGB color space in $J(x)$. $\Omega(x)$ is a neighborhood centered on pixel X in image $J(x)$. The image defogging algorithm based on dark channel prior includes the following steps:

1. Estimate and refine the transmittance value. The idea is as follows: Assuming that the atmospheric care A is known, and $A>0$, divide both the left and right sides by A^c to get Equation (5):

$$\frac{I^c(x)}{A^c} = t(x)\frac{J^c(x)}{A^c} + 1 - t(x), c \in (r, g, b). \quad (5)$$

   By minimizing the above formula, you can get:

$$\min_{y\in\Omega(x)}(\min_{c\in(r,g,b)}\frac{I^c(y)}{A^c}) = \tilde{t}(x)\min_{y\in\Omega(x)}(\min_{c\in(r,g,b,)}\frac{J^c(y)}{A^c}) + 1 - \tilde{t}(x). \quad (6)$$

   In Equation (6), it is constant only in a small neighborhood, so it is not needed to minimize it. And the dark channel in the clear image is 0. The predicted transmittance can be corrected by introducing a factor w between 0 and 1 (typically 0.95) to make the defogged image more natural.
2. Estimate the atmospheric illumination value.
3. Substituting the transmittance value $t(x)$ and atmospheric illumination value A obtained in Equation (1) into Equation (2), and obtaining the defogged image of the input image.
4. The image that needs to be defogged is restored by a dark channel prior algorithm based on guided filtering. After the image is restored, considering that the color quality of the image is a little black, the final image is obtained by brightness enhancement.

3.3 UDP-YOLO Network Structure

3.3.1 Improved Multi-Scale Prediction Network

For the application of autonomous driving technology in foggy scenes, there are usually small objects such as people and bicycles in the scenes, while YOLOv4-tiny model has only two prediction scales of 13×13 and 26×26. Using YOLOv4-tiny model to detect the data set we selected in this paper, it is found that the detection effect of small objects such as people and bicycles is poor. Therefore, inspired by FPN, PANet and NAS-FPN, this paper proposes a new neck network for feature fusion without adding too many model parameters, and adds a small target detection head. The improved network structure is shown in Figure 4.

As can be seen from Figure 4, this paper has improved the original MobileNetv1 network, and made the following modifications to the original stage1, stage2, and stage3, reducing the number of convolution kernels of stage1 and making it output a feature map with a size of 64×64. Then, CSP module was added between stage1 and stage2 to make it input a feature map with a size of 128×128 and extract it. Stage2 is also modified to output and extract the feature map with the size of 256×256. Finally, the number of convolution kernels of stage3 is trimmed to output and extract the feature map with the size of 512×512. The characteristic graphs of these three dimensions are respectively marked as F1, F2 and F3. Next, feature fusion is performed. Firstly, F3 is convolved and downsampled to get F3.1, and F2 is convolved to get F2.1. F1 is convolved and upsampled to get F1.1, and then it is fused to get our first fused feature. After that, F1.1 is up-sampled, and fused with the feature map of F3 after convolution operation to obtain the second feature. At last, F2.1 is downsampled to get F2.2, and it is fused with the feature map obtained by convolution operation of F1 to get the third feature. These three fusion features not only contain strong detail information, but also have great semantic information, so the detected results are very comprehensive.

3.3.2 Expand Receptive Field

Expanding receptive field in the model, a low-cost measure, is helpful to improve the feature extraction ability, thus improving the performance. In this paper, we put the PPM module into the feature maps of P3, P4 and P5 extracted from the improved YOLOv4-tiny model in Figure 4, so as to increase the feature extraction ability. The PPM module can divide the extracted feature maps into two branches, one of which is divided into multiple sub-areas for GAP (Global Average Pooling) operation, then adjust the channel size through convolution operation, and then obtain the unpooled feature map through bilinear interpolation. The PPM module [58] consists of five steps:

1. Pool the feature map extracted from the backbone network to obtain a feature pyramid.
2. Get the characteristic maps with the size of 1×1, 2×2, 3×3, 6×6 and channel $= 1/N$ through the 1×1 depth convolution descending channel.
3. Bilinear interpolation filling and upsampling the feature map to the original feature map size.
4. Channel splicing with the feature map to obtain a feature map with double channel number.
5. Using 1×1 convolution kernel to deeply convolve and channel down the spliced feature map to obtain the final prediction result which is consistent with the channel number of the input feature map.

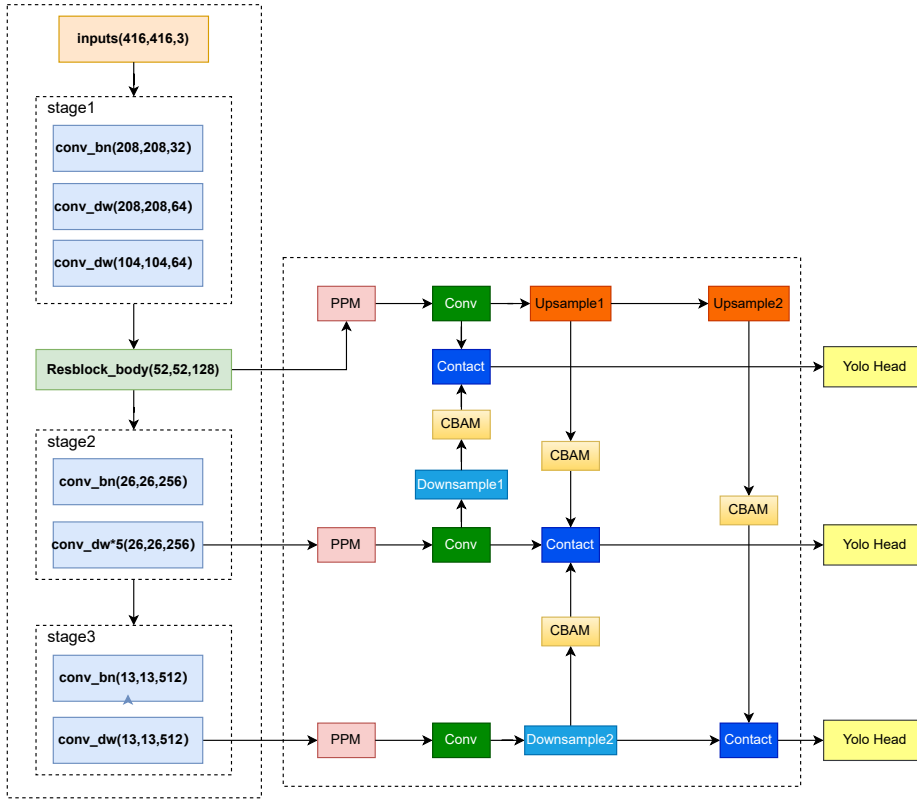


Figure 4. UDP-YOLO model structure

3.3.3 Attention Mechanism

Deep Convolutional Neural Network (CNN) has been widely used in computer field, and has made great progress in image recognition, object detection and semantic segmentation. Because the performance of the original model will be slightly degraded by pruning, this paper adds a lightweight attention module CBAM module while considering the speed and performance. This module can conduct attention mechanism in space and channel, deduce the attention weight coefficient along the channel and space dimensions, and then multiply it with feature map to adjust the features adaptively. Because CBAM is a lightweight general-purpose module, it can be seamlessly integrated into any CNN architecture, and its computational cost is basically negligible. And can carry out end-to-end training with basic CNN. On different classification and detection data sets, after integrating CBAM into different models, the performance of the models has been consistently improved, showing its wide applicability.

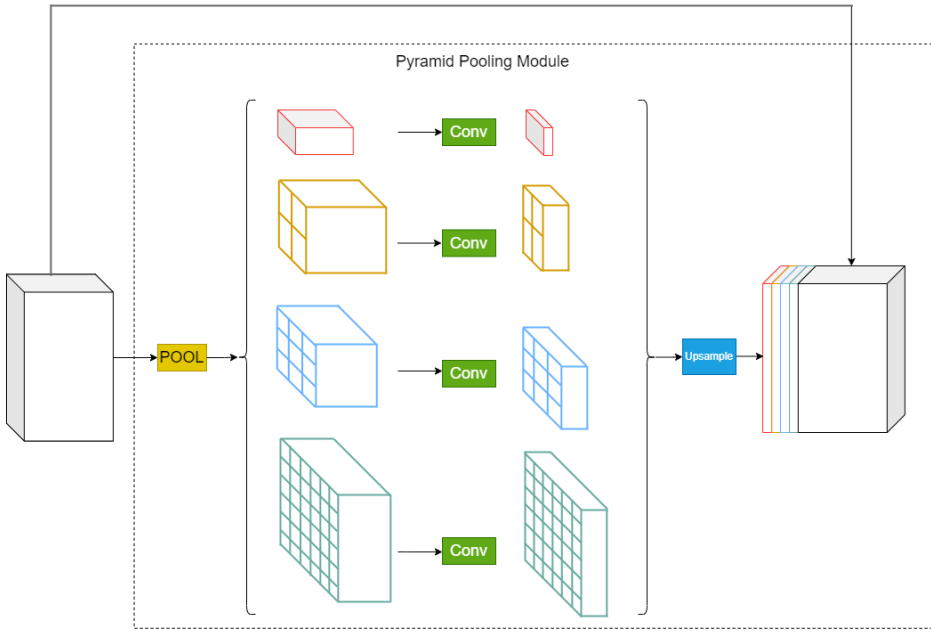


Figure 5. The module of PPM

As shown in Figure 6, CBAM module is divided into two sections: channel attention module and spatial attention module. This module can not only save parameters and attention, but also ensure that it can be integrated into the existing network architecture as a plug-and-play module.

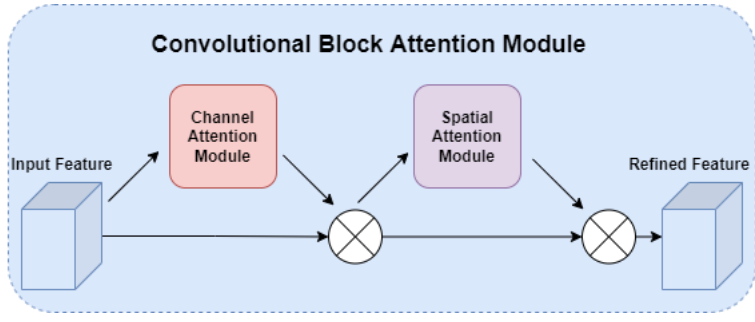


Figure 6. The module of CBAM

The CAM module is shown in Figure 7, the attention module on the channel firstly passes the input feature map F through global max pooling and global average pooling based on width and height respectively to obtain two $1 \times 1 \times C$ feature

maps C1 and C2. C1 will be upsampled. Then C1 and C2 respectively pass through a neural network sharing two layers, the number of neurons in the first layer is C/r (r is the reduction rate), and the number of neurons in the second layer is C. The features of MLP output are added based on element-wise, and then the final channel attention feature, namely M_C, which is generated by LA (LeakyRelu Activation) operation. Finally, M_C and input F are multiplied by element-wise, and the required input features of the next module are obtained.

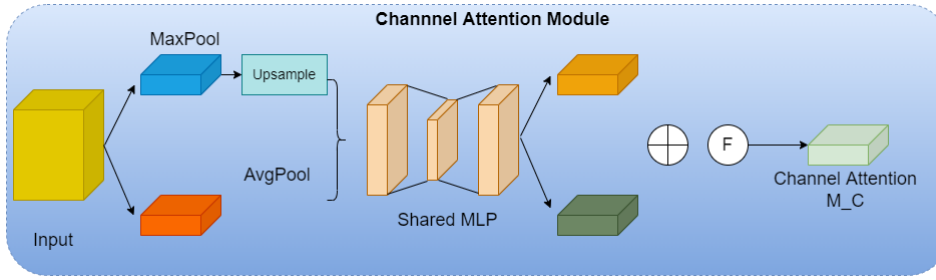


Figure 7. The module of CAM

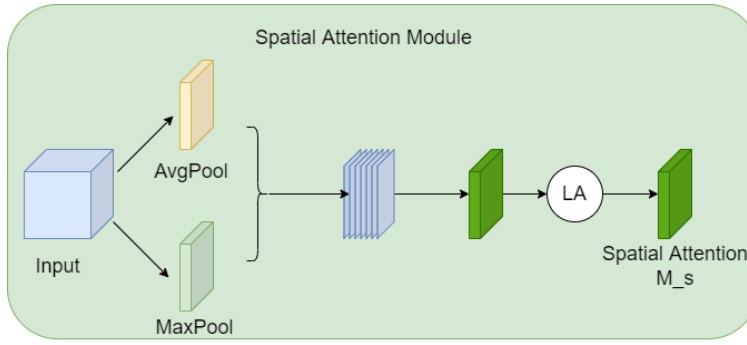


Figure 8. The module of SAM

As shown in Figure 8, SAM module uses the features obtained in the last round as a global max pooling and a global average pooling based on channel to obtain two $H \times W \times 1$ feature maps, then performs channel splicing operation on these two feature maps based on channel, and then reduces the dimension to one channel through a 7×7 convolution operation. Then the Leaky ReLU generates the spatial attention feature, namely M_s. Finally, the feature and the original input features of the module are multiplied to obtain the final features for detection.

4 EXPERIMENT

4.1 Experimental Data and Experimental Platform

The data sets adopted in this paper are Foggy-cityscape data set and BDD100k data set. Foggy-cityscape data set is a data set obtained by photographing the road conditions of many foreign cities, which can be used for object detection and segmentation. BDD100K data set is the largest and most diverse open driving data set published by Berkeley Artificial Intelligence Laboratory. In this work, a part of BDD100K data set and Foggy-cityscape data set are selected for fusion experiment, and the fused data set is divided into training set for experiment according to the proportion, and the test set is used as evaluation. In this paper, mAP and FPS are used to evaluate the performance of the model. Table 1 describes the details of the data set we selected.

	Foggy-cityscape and BDD100k
Number of classes	6
Training datasets	5 976
Test datasets	960

Table 1. Datasets details

The operating system used in this experimental platform is Windows 10, the processor is Intel (R) Core (TM) i7-4790 KCPU @ 4.00 GHz, the running memory is 32 GB, the GPU is NVIDIA GeForce GTX 3060, and the parallel computing framework version is CUDA 11.6.

The flow chart of this experiment is shown in Figure 9. Firstly, select the desired dataset from the Foggy-cityscape dataset and the BDD100k dataset, then discriminate the selected data set by fog judgment algorithm, defog the foggy data set, and finally fuse the defogged data set and the fog-free data set to divide the training set and the test set, and carry out adaptive brightness enhancement. Finally, the original YOLOv4-tiny model and our proposed UDP-YOLO model are used for detection. Figure 10 shows the effect diagram after each step of operation.

4.2 Training Model

After the image processing of the data set pair, we start to train the processed data set. During the training, we adopt the default size of YOLOv4-tiny (416, 416), and set the input batch size to 8 and the momentum to 0.9. Firstly, without using the pre-training weight, only the backbone network is loaded for training to get a better weight. Then, this weight is put into the model as the pre-training weight to train 50 epochs, and then 250 epochs are trained to get the performance of our original YOLOv4-tiny model test data set. During the pre-training, the learning rate of our first 50 training sessions was set at 0.001, and then the learning rate was gradually reduced from 0.001 to 0.0001 by adopting cosine annealing. Then we

add our improved structure in turn, and take the weight of the best performance measured last time as the pre-training weight for training.

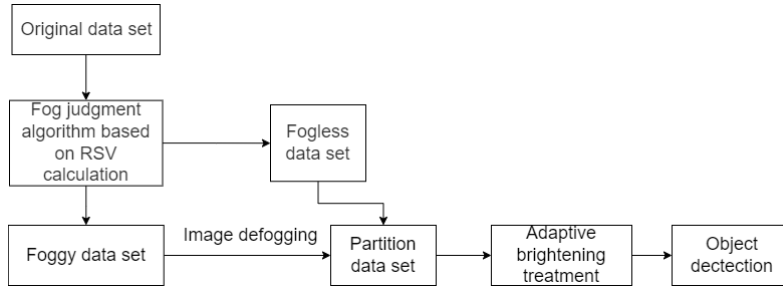


Figure 9. Experimental flow chart

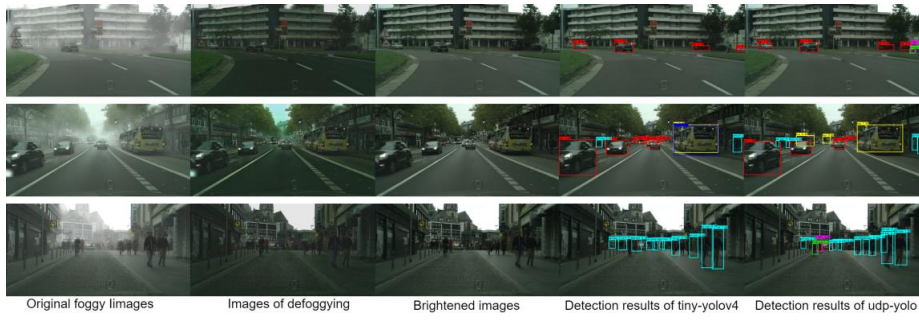


Figure 10. Image processing and target detection process

4.3 Ablation Experiment

To prove the effectiveness of our improved model, we first processed the original data through image processing, and detected the processed data by using YOLOv4-tiny model. By replacing the backbone network with MobileNetv1 and reducing the network parameters, we improved the neck network to form a new multi-feature fusion structure. Add PPM module and CBAM module in turn. The validity of the improved model is verified by mAP, FPS.

As shown in Table 2, after replacing the backbone network from CSPDarknet-tiny with MobileNetv1, and greatly modifying the neck network to form a new feature fusion structure, our average performance of detecting six types of objects has increased from 19.75 % to 32.41 %, but its speed has also decreased from the

	YOLOv4-tiny	UDP-YOLO				
MobileNetv1	–	+	+	+	+	+
Delete parameter	–	–	+	+	+	+
CSP–MobileNetv1	–	–	–	+	+	+
Add PPM	–	–	–	–	+	+
Add CBAM	–	–	–	–	–	+
mAP (%)	19.36	32.41	31.74	34.27	36.86	40.54
FPS	100.3	78.3	90.2	83.1	68.5	61.7

Table 2. The result of Ablation experiment. The '–' sign indicates that the operation corresponding to the first column of the table was not performed in the ablation experiment, and the '+' sign indicates that the operation corresponding to the first column of the table was performed in the ablation experiment.

original 100.3 to 78.3. Therefore, in order to maintain its speed, the convolution kernel of the backbone network is pruned. After pruning, the observation results show that while the performance decreases by less than one point, our speed increases by about 15 %. After that, we added CSP module to the backbone network to form CSP-MobileNet, and the detection performance increased from 31.74 % to 34.27 % when the speed decreased by less than 8 %. After that, by adding PPM module to increase the receptive field of our model, the performance is improved to 36.86 % and the speed is reduced to 68.5. Finally, we added a lightweight attention module CBAM to the model, which improved our performance to 40.54 % and the speed to 61.7.

4.4 Comparison of UDP-YOLO and YOLO Series Models

In this section, we compare the performance of the non-lightweight model of YOLO series with that of UDP-YOLO model proposed in this paper, because the performance of YOLOv4-tiny model in detecting the data set selected in this paper is low. YOLOv3 is the third version of YOLO series. The test result of this model on the selected data set is 41.35 %, but its FPS is only 48.6. Compared with UDP-YOLO proposed in this paper, its performance is 0.81 % higher, but its speed is much lower. After that, this paper replaces the backbone network of YOLOv3, Darknet53, with EfficientNet, and finds that its performance and speed are not as good as YOLOv3. YOLOv4 is based on YOLOv3, and the measured performance of this model is 42.85 %, which is 3.6 % higher than YOLOv3 and 5.6 % higher than UDP-YOLO. But the speed is reduced by 31.7 % compared with UDP-YOLO. MobileNetv2-YOLOv4 replaces the CSP-Darknet53 backbone network of YOLOv4 with MobileNetv2 for training, and evaluates the best weight after training. Although compared with YOLOV4 in speed, it is still not as fast as UDP-YOLO proposed in this paper. And its performance has been greatly reduced.

From Figure 3, we can see that. Compared with the model of YOLO series, the effect of replacing the backbone network of YOLO series and detecting it is not

	YOLOv3	EfficientNet-YOLOv3	YOLOv4	MobileNetv2-YOLOv4	UDP-YOLO
mAP (%)	41.35	27.68	42.85	28.05	40.54
FPS	48.6	46.5	42.3	52.1	61.7

Table 3. Comparison of UDP-YOLO and YOLO series models

as good as the original effect. However, the UDP-YOLO proposed in this paper, although the performance of the backbone network is a little reduced after it is replaced by a new network, is indeed much faster. In fact, we can completely increase the performance by reducing the speed, but this principle is not adopted due to the real-time requirement of autonomous driving technology. The comparison of mAP and FPS shows that the UDP-YOLO model proposed by us is completely feasible.

4.5 Comparison of UDP-YOLO and Other Lightweight Models

In this section, we use MobileNetv2-SSD, YOLOv5s, YOLOx-tiny and EfficientNet to test our selected data set, and compare the performance with our UDP-YOLO model. The experimental environment and details are the same as before.

	MobileNetv2-SSD	YOLOv5s	YOLOx-tiny	EfficientNet	UDP-YOLO
mAP (%)	22.91	38.50	37.56	34.33	40.54
FPS	75.5	25.3	48.5	30.1	61.7

Table 4. Comparison of UDP-YOLO and other lightweight models

As can be seen from Table 4, the UDP-YOLO model is at the optimal performance compared to all four of these networks. Compared to the MobileNetv2-SSD model, its performance almost doubles, although its speed is reduced by about 22 %, which is a good indication of the efficiency of our model.

4.6 Comparison Experiments Using Our Model on Road Defect Dataset

In autonomous driving technology, in addition to avoiding vehicles travelling in the road, self-driving cars also need to avoid some road defects by predicting them in advance. To demonstrate the efficient generalisation of the model proposed in this paper, experiments are conducted on the GRDDC2020 road defect detection dataset using the UDP-YOLO model. Table 5 shows the experiments comparing our proposed model with some other lightweight models.

By looking at Table 5, we see that the performance of our proposed model is better than other models, and we can show after these experiments that our proposed model is effective for the application of autonomous driving technology in this direction of computer vision.

	YOLOv4-tiny	YOLOv4	YOLOv5s	Tiny-YOLOX	UDP-YOLO
mAP (%)	52.45	54.56	56.79	56.95	57.30
FPS	101.5	30.6	24.8	45.3	62.8

Table 5. Comparison experiments using our model on road defect dataset

5 CONCLUSIONS

In this work, we propose an improved model based on YOLOv4-tiny to deal with the problem of avoiding vehicles while driving, and to demonstrate the effectiveness of our proposed model, we also carry out generalisation experiments in case of road defects. The evaluation metrics of our experimental results show that our model outperforms these comparative lightweight models, allowing us to detect the target briskly. However, there is still room for improvement in our proposed model. For example, the performance of our detection is not robust enough, which may prevent us from avoiding a vehicle in an accident while driving because we do not detect the target. Therefore, we will continue working on this issue and further to improve and experiment with the model to come up with a more efficient model that detects the target in the shortest possible time.

Acknowledgements

This research was supported by the National Natural Science Foundation of China under the project "Research on Theory and Control Protocols of Converged Multiple Access Communication Networks" (No. 61461053), as well as the National Innovation and Entrepreneurship Program for College Students (No. 202210673062) and the expert workstation of Ding Hongwei.

Author Contributions

- Yonghao Liu: Proposal of the method, experimentation, writing;
- Hongwei Ding: Supervision, revision of the original;
- Zhijun Yang: Supervision, validation;
- Guangen Ding: Data collection;
- Peng Hu: Investment, supervision;
- Qianxue Xu: Data collection.

REFERENCES

[1] Lian, J.—Yang, Z.—Liu, J.—Sun, W.—Zheng, L.—Du, X.—Yi, Z.—Shi, B.—Ma, Y.: An Overview of Image Segmentation Based on Pulse-Coupled

Neural Network. Archives of Computational Methods in Engineering, Vol. 28, 2021, No. 2, pp. 387–403, doi: 10.1007/s11831-019-09381-5.

[2] Li, J.—Zhang, C.—Zhou, J. T.—Fu, H.—Xia, S.—Hu, Q.: Deep-LIFT: Deep Label-Specific Feature Learning for Image Annotation. IEEE Transactions on Cybernetics, Vol. 52, 2021, No. 8, pp. 7732–7741, doi: 10.1109/TCYB.2021.3049630.

[3] Liu, S.—Liu, D.—Srivastava, G.—Połap, D.—Woźniak, M.: Overview and Methods of Correlation Filter Algorithms in Object Tracking. Complex & Intelligent Systems, Vol. 7, 2021, No. 4, pp. 1895–1917, doi: 10.1007/s40747-020-00161-4.

[4] Brunetti, A.—Buongiorno, D.—Trotta, G. F.—Bevilacqua, V.: Computer Vision and Deep Learning Techniques for Pedestrian Detection and Tracking: A Survey. Neurocomputing, Vol. 300, 2018, pp. 17–33, doi: 10.1016/j.neucom.2018.01.092.

[5] Minaee, S.—Luo, P.—Lin, Z.—Bowyer, K.: Going Deeper into Face Detection: A Survey. CoRR, 2021, doi: 10.48550/arXiv.2103.14983.

[6] Zhu, Y.—Du, J.: TextMountain: Accurate Scene Text Detection via Instance Segmentation. Pattern Recognition, Vol. 110, 2021, Art. No. 107336, doi: 10.1016/j.patcog.2020.107336.

[7] Liu, R. W.—Guo, Y.—Lu, Y.—Chui, K. T.—Gupta, B. B.: Deep Network-Enabled Haze Visibility Enhancement for Visual IoT-Driven Intelligent Transportation Systems. IEEE Transactions on Industrial Informatics, Vol. 19, 2023, No. 2, pp. 1581–1591, doi: 10.1109/TII.2022.3170594.

[8] Cheng, G.—Si, Y.—Hong, H.—Yao, X.—Guo, L.: Cross-Scale Feature Fusion for Object Detection in Optical Remote Sensing Images. IEEE Geoscience and Remote Sensing Letters, Vol. 18, 2021, No. 3, pp. 431–435, doi: 10.1109/LGRS.2020.2975541.

[9] Girshick, R.—Donahue, J.—Darrell, T.—Malik, J.: Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 580–587, doi: 10.1109/CVPR.2014.81.

[10] Girshick, R.: Fast R-CNN. Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.

[11] Ren, S.—He, K.—Girshick, R.—Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 39, 2017, No. 6, pp. 1137–1149, doi: 10.1109/TPAMI.2016.2577031.

[12] He, K.—Gkioxari, G.—Dollár, P.—Girshick, R.: Mask R-CNN. Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980–2988, doi: 10.1109/ICCV.2017.322.

[13] Redmon, J.—Divvala, S.—Girshick, R.—Farhadi, A.: You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[14] Redmon, J.—Farhadi, A.: YOLO9000: Better, Faster, Stronger. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 7263–7271, doi: 10.1109/CVPR.2017.690.

[15] Redmon, J.—Farhadi, A.: YOLOv3: An Incremental Improvement. CoRR, 2018, doi: 10.48550/arxiv.1804.02767.

[16] Bochkovskiy, A.—Wang, C. Y.—Liao, H. Y. M.: YOLOv4: Optimal Speed and Accuracy of Object Detection. CoRR, 2020, doi: 10.48550/arXiv.2004.10934.

[17] Wang, C. Y.—Bochkovskiy, A.—Liao, H. Y. M.: Scaled-YOLOv4: Scaling Cross Stage Partial Network. Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 13029–13038, doi: 10.1109/CVPR46437.2021.01283.

[18] Liu, W.—Anguelov, D.—Erhan, D.—Szegedy, C.—Reed, S.—Fu, C. Y.—Berg, A. C.: SSD: Single Shot Multibox Detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (Eds.): Computer Vision – ECCV 2016. Springer, Cham, Lecture Notes in Computer Science, Vol. 9905, 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.

[19] Jiang, Z.—Zhao, L.—Li, S.—Jia, Y.: Real-Time Object Detection Method Based on Improved YOLOv4-Tiny. CoRR, 2020, doi: 10.48550/arXiv.2011.04244.

[20] Chen, D.—He, M.—Fan, Q.—Liao, J.—Zhang, L.—Hou, D.—Yuan, L.—Hua, G.: Gated Context Aggregation Network for Image Dehazing and Deraining. 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), 2019, pp. 1375–1383, doi: 10.1109/WACV.2019.00151.

[21] Wu, H.—Tan, Z.: An Image Dehazing Algorithm Based on Single-Scale Retinex and Homomorphic Filtering. In: Liang, Q., Wang, W., Liu, X., Na, Z., Jia, M., Zhang, B. (Eds.): Communications, Signal Processing, and Systems (CSPS 2019). Springer, Singapore, Lecture Notes in Electrical Engineering, Vol. 571, 2020, pp. 1482–1493, doi: 10.1007/978-981-13-9409-6_178.

[22] Wang, L. J.—Zhu, R.: Image Defogging Algorithm of Single Color Image Based on Wavelet Transform and Histogram Equalization. Applied Mathematical Sciences, Vol. 7, 2013, No. 79, pp. 3913–3921, doi: 10.12988/ams.2013.34206.

[23] Fan, T.—Li, C.—Ma, X.—Chen, Z.—Zhang, X.—Chen, L.: An Improved Single Image Defogging Method Based on Retinex. 2017 2nd International Conference on Image, Vision and Computing (ICIVC), IEEE, 2017, pp. 410–413, doi: 10.1109/ICIVC.2017.7984588.

[24] Cai, B.—Xu, X.—Jia, K.—Qing, C.—Tao, D.: DehazeNet: An End-to-End System for Single Image Haze Removal. IEEE Transactions on Image Processing, Vol. 25, 2016, No. 11, pp. 5187–5198, doi: 10.1109/TIP.2016.2598681.

[25] Li, B.—Peng, X.—Wang, Z.—Xu, J.—Feng, D.: AOD-Net: All-in-One Dehazing Network. Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 4770–4778, doi: 10.1109/ICCV.2017.511.

[26] Jeong, K.—Choi, K.—Kim, D.—Song, B. C.: Fast Fog Detection for De-Fogging of Road Driving Images. IEICE Transactions on Information and Systems, Vol. E101.D, 2018, No. 2, pp. 473–480, doi: 10.1587/transinf.2017EDP7211.

[27] Liang, Y. T.—Li, L.—Zhao, K. B.—Hu, J. H.: Defogging Algorithm of Color Images Based on Gaussian Function Weighted Histogram Specification. 2016 10th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), IEEE, 2016, pp. 364–369, doi: 10.1109/SKIMA.2016.7916248.

[28] Xu, H.—Guo, J.—Liu, Q.—Ye, L.: Fast Image Dehazing Using Improved Dark

Channel Prior. 2012 IEEE International Conference on Information Science and Technology, 2012, pp. 663–667, doi: 10.1109/ICIST.2012.6221729.

[29] Jiang, X.—Yao, H.—Zhang, S.—Lu, X.—Zeng, W.: Night Video Enhancement Using Improved Dark Channel Prior. 2013 IEEE International Conference on Image Processing, 2013, pp. 553–557, doi: 10.1109/ICIP.2013.6738114.

[30] Fu, Z.—Yang, Y.—Shu, C.—Li, Y.—Wu, H.—Xu, J.: Improved Single Image Dehazing Using Dark Channel Prior. Journal of Systems Engineering and Electronics, Vol. 26, 2015, No. 5, pp. 1070–1079, doi: 10.1109/JSEE.2015.00116.

[31] Viola, P.—Jones, M.: Rapid Object Detection Using a Boosted Cascade of Simple Features. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), Vol. 1, 2001, doi: 10.1109/CVPR.2001.990517.

[32] Dalal, N.—Triggs, B.: Histograms of Oriented Gradients for Human Detection. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05), Vol. 1, 2005, pp. 886–893, doi: 10.1109/CVPR.2005.177.

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

[34] Felzenszwalb, P. F.—Girshick, R. B.—McAllester, D.—Ramanan, D.: Object Detection with Discriminatively Trained Part-Based Models. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 32, 2010, No. 9, pp. 1627–1645, doi: 10.1109/TPAMI.2009.167.

[35] Zaidi, S. S. A.—Ansari, M. S.—Aslam, A.—Kanwal, N.—Asghar, M.—Lee, B.: A Survey of Modern Deep Learning Based Object Detection Models. Digital Signal Processing, Vol. 126, 2022, Art. No. 103514, doi: 10.1016/j.dsp.2022.103514.

[36] He, K.—Zhang, X.—Ren, S.—Sun, J.: Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 37, 2015, No. 9, pp. 1904–1916, doi: 10.1109/TPAMI.2015.2389824.

[37] Lin, T. Y.—Dollár, P.—Girshick, R.—He, K.—Hariharan, B.—Belongie, S.: Feature Pyramid Networks for Object Detection. Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2117–2125, doi: 10.1109/CVPR.2017.106.

[38] Dai, J.—Li, Y.—He, K.—Sun, J.: R-FCN: Object Detection via Region-Based Fully Convolutional Networks. In: Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., Garnett, R. (Eds.): Advances in Neural Information Processing Systems 29 (NIPS 2016). Curran Associates, Inc., 2016, pp. 397–387.

[39] Qiao, S.—Chen, L. C.—Yuille, A.: Detectors: Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution. Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 10213–10224, doi: 10.1109/CVPR46437.2021.01008.

[40] Lin, T. Y.—Goyal, P.—Girshick, R.—He, K.—Dollár, P.: Focal Loss for Dense Object Detection. 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2999–3007, doi: 10.1109/ICCV.2017.324.

[41] Duan, K.—Bai, S.—Xie, L.—Qi, H.—Huang, Q.—Tian, Q.: CenterNet: Keypoint Triplets for Object Detection. Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 6568–6577, doi: 10.1109/ICCV.2019.00667.

[42] Tan, M.—Pang, R.—Le, Q. V.: EfficientDet: Scalable and Efficient Object Detection. Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 10778–10787, doi: 10.1109/CVPR42600.2020.01079.

[43] Liu, S.—Qi, L.—Qin, H.—Shi, J.—Jia, J.: Path Aggregation Network for Instance Segmentation. Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 8759–8768, doi: 10.1109/CVPR.2018.00913.

[44] Ghiasi, G.—Lin, T. Y.—Le, Q. V.: NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection. Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 7029–7038, doi: 10.1109/CVPR.2019.00720.

[45] Hu, J.—Shen, L.—Sun, G.: Squeeze-and-Excitation Networks. Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141, doi: 10.1109/CVPR.2018.00745.

[46] Wang, Q.—Wu, B.—Zhu, P.—Li, P.—Zuo, W.—Hu, Q.: ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 11531–11539, doi: 10.1109/CVPR42600.2020.01155.

[47] Li, X.—Wang, W.—Hu, X.—Yang, J.: Selective Kernel Networks. Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 510–519, doi: 10.1109/CVPR.2019.00060.

[48] Woo, S.—Park, J.—Lee, J. Y.—Kweon, I. S.: CBAM: Convolutional Block Attention Module. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (Eds.): Computer Vision – ECCV 2018. Springer, Cham, Lecture Notes in Computer Science, Vol. 11211, 2018, pp. 3–19, doi: 10.1007/978-3-030-01234-2_1.

[49] Fu, J.—Liu, J.—Tian, H.—Li, Y.—Bao, Y.—Fang, Z.—Lu, H.: Dual Attention Network for Scene Segmentation. Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 3141–3149, doi: 10.1109/CVPR.2019.00326.

[50] He, Y.—Zhang, X.—Sun, J.: Channel Pruning for Accelerating Very Deep Neural Networks. Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 1398–1406, doi: 10.1109/ICCV.2017.155.

[51] Zhu, M.—Gupta, S.: To Prune, or Not to Prune: Exploring the Efficacy of Pruning for Model Compression. CoRR, 2017, doi: 10.48550/arXiv.1710.01878.

[52] Kwon, S. J.—Lee, D.—Kim, B.—Kapoor, P.—Park, B.—Wei, G. Y.: Structured Compression by Weight Encryption for Unstructured Pruning and Quantization. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 1906–1915, doi: 10.1109/CVPR42600.2020.00198.

[53] Anwar, S.—Hwang, K.—Sung, W.: Structured Pruning of Deep Convolutional Neural Networks. ACM Journal on Emerging Technologies in Computing Systems

(JETC), Vol. 13, 2017, No. 3, Art. No. 32, doi: 10.1145/3005348.

[54] Tan, Z.—Song, J.—Ma, X.—Tan, S. H.—Chen, H.—Miao, Y.—Wu, Y.—Ye, S.—Wang, Y.—Li, D.—Ma, K.: PCNN: Pattern-Based Fine-Grained Regular Pruning Towards Optimizing CNN Accelerators. 2020 57th ACM/IEEE Design Automation Conference (DAC), 2020, pp. 1–6, doi: 10.1109/DAC18072.2020.9218498.

[55] de Kruif, B. J.—de Vries, T. J. A.: Pruning Error Minimization in Least Squares Support Vector Machines. IEEE Transactions on Neural Networks, Vol. 14, 2003, No. 3, pp. 696–702, doi: 10.1109/TNN.2003.810597.

[56] Yang, M.—Faraj, M.—Hussein, A.—Gaudet, V.: Efficient Hardware Realization of Convolutional Neural Networks Using Intra-Kernel Regular Pruning. 2018 IEEE 48th International Symposium on Multiple-Valued Logic (ISMVL), 2018, pp. 180–185, doi: 10.1109/ISMVL.2018.00039.

[57] He, Y.—Kang, G.—Dong, X.—Fu, Y.—Yang, Y.: Soft Filter Pruning for Accelerating Deep Convolutional Neural Networks. CoRR, 2018, doi: 10.48550/arXiv.1808.06866.

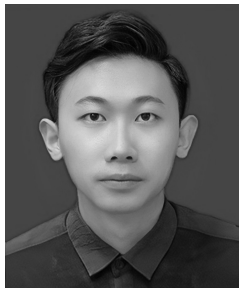
[58] Zhao, H.—Shi, J.—Qi, X.—Wang, X.—Jia, J.: Pyramid Scene Parsing Network. Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6230–6239, doi: 10.1109/CVPR.2017.660.

Yonghao Liu received his Master degree in the Department of Communication and Information System, Yunnan University. He is currently pursuing his Ph.D. in the School of Computer Science and Technology, HUST. His research interests include image processing and neural network.

Hongwei Ding is Professor and Ph.D. Supervisor with the Yunnan University. He is mainly engaged in deep reinforcement learning and generative adversarial networks and has published many scientific papers indexed by SCI and EI.

Zhijun Yang is External Professor and Master's Tutor of the Yunnan University, mainly engaged in deep reinforcement learning and generative adversarial network research, and has published several academic papers indexed by SCI and EI.

Qianxue Xu is currently pursuing her B.Sc. degree in communication engineering at the Yunnan University. Her research interests include deep learning, image processing and object detection.



Guangen Ding is currently pursuing his M.Sc. degree in real estate management at the Yunnan University of Finance and Economics. His research interests include deep learning, building crack detection.

Peng Hu is working in the R & D Department of Kunming Ubay Technology Co. in China. His research interests include communication and information systems, deep learning and object tracking.

Computing and Informatics, Vol. 42, 2023, 861–877, doi: 10.31577/cai_2023_4_861

EXPERT MINING COLLABORATIVE FILTERING RECOMMENDATION ALGORITHM BASED ON SIGNAL FLUCTUATION

Shuo Wang, Jing Yang*, Fanshu Shang, Jingyun Sun

Harbin Engineering University
College of Computer Science and Technology
No. 145, Nantong Street, Nangang District
Harbin City, Heilongjiang Province, China
e-mail: yangjing@hrbeu.edu.cn

Abstract. This paper proposes an advanced expert collaborative filtering recommendation algorithm. Although ordinary expert system filtering algorithms have improved the recommendation accuracy of collaborative filtering technology to a certain extent, they have not screened the level of expertise of experts, and the credibility of experts varies. Therefore, this paper proposes an expert mining system based on signal fluctuations. The algorithm uses signal processing technology to filter the level of experts. This method introduces a kurtosis factor. Regarding the user's rating sequence as a random discrete signal, and then randomly sorting the user's ratings k times, the average kurtosis of the user is obtained. And take the kurtosis value as the credibility of expert users. Through experiments on multiple datasets including MovieLens, Jester, Booking-Crossings, and Last.fm, we have proved the advancement and reliability of our method.

Keywords: Recommendation system, machine learning, expert system, kurtosis, collaborative filtering

1 INTRODUCTION

With the advent of the era of big data, information technology has developed rapidly, and data has grown explosively. How to quickly and effectively obtain valuable in-

* Corresponding author

formation from complex data has become a key issue in the current development of big data. Recommendation algorithm [1] as one of the most effective means to solve the "information overload" [2], it is now not only widely used in daily life such as shopping, social networking, entertainment and other platforms, but even in medical treatment, education, etc. There is also rapid progress in the field. It can effectively use existing resources to filter out valuable information from massive data and feed it back to users. In artificial intelligence, an expert system is a computer program that replicates the ability for evaluation of a human expert. Expert systems are created to reason through knowledge bases that are mostly expressed as if-then rules, as opposed to using conventional procedural code. A knowledge base, a search or inference engine, a knowledge acquisition system, and a user interface or communication system are the four main parts of an expert system. Knowledge systems execute inference operations using explicitly stated knowledge to solve challenging challenges in the real world.

In the early phases of development, a typical recommendation approach is to simply arrange goods according to sales, topic clicks, or news reading, etc., before selecting the top N items to generate a ranking list and suggest them to users. This technique produces excellent results, and many websites still use comparable features today. However, this approach also has a serious drawback in that it can only recommend a small number of highly ranked items and cannot harvest more long-tail data. Therefore, the primary objective of study in the field of recommendation systems has become how to fully utilize current resources (items) to produce recommendations that are as accurate and thorough as feasible. At present, many papers have conducted in-depth research on recommendation systems and proposed many recommendation algorithms. Among many recommendation algorithms, collaborative filtering algorithm [3, 4, 5, 6], as the earliest and most successful recommendation technology, is the mainstream research direction in the field of recommendation systems. Its task is to use the scoring matrix of users and items to predict high-scoring items and recommend them to users. It first recommends user entry items to target users based on attributes. Recommender systems that recommend things based on consumer collaborations are the most extensively used and validated technique of giving recommendations. User-to-user collaborative filtering and item-to-item collaborative filtering are the two forms, both of which are based on user-to-user similarity. To solve some of the drawbacks of content-based filtering, collaborative filtering provides recommendations based on similarities between users and items simultaneously.

Although the traditional collaborative filtering method is effective, it is not reliable enough [7]. In social psychology, there is a principle of authority: that is, authority has a powerful force that can affect people's behavior, and people are more willing to listen to the opinions of experts. Users must accept their employees' opinion with respect and gratitude, even if they disagree, if they want to effectively impact their employees' performance utilizing the principle of authority. This demonstrates to users' staff that users are paying attention and that they are free to share their own knowledge. Therefore, many scholars have carried out re-

search on integrating expert opinions into recommendation systems. In 2016, Hwang et al. [8] proposed a method combining category experts and collaborative filtering technology-CE method. This method selects a small number of users as experts in each category, and replaces their scores with those of ordinary neighbors' scores. Although the recommendation accuracy of collaborative filtering technology has been improved to a certain extent, this method does not screen the level of expertise of experts, and the credibility of experts varies. Therefore, this paper proposes an expert mining collaborative filtering recommendation algorithm based on signal fluctuation method, which uses signal processing technology. In general, signal fluctuation reduces system performance when compared to nonfluctuating signals. This loss is significantly reduced when there is perfect independence between subsequent signals and rather prominent when there is total correlation between signals. Section 2 of this paper introduces related work; Section 3 gives a detailed description of the method proposed in this paper. Section 4 gives the experimental settings and description of the dataset. Section 5 gives the experimental results and analysis of the experimental results. Section 6 analyzes the time complexity. Section 7 is the conclusion and future work.

2 RELATED WORK

As an important part of the recommendation system, the collaborative filtering (CF) algorithm has received extensive attention from industry and academia. The collaborative filtering algorithm is based on a strong presupposition: if it is observed that a user has consumed item A, then there is a high probability that the user will like item B similar to A, and similar users will likely like the same one entry [9]. Therefore, the core of collaborative filtering is to describe the similarity between items and users, and use the behavior of users similar to the users to be recommended to infer the preference of the users to be recommended for a particular product, and then make corresponding recommendations based on this preference. Currently, collaborative filtering algorithms mainly include two types: model-based and neighbor-based. Model-based algorithms learn prediction models from known scores, which have obvious advantages in improving prediction accuracy and coping with data sparsity. Collaborative filtering is a subset of models used in recommendation systems that examines for patterns between users or between objects using ratings or preferences that have been collected for both the person and the item. Neighborhood-based collaborative filtering algorithms, also known as memory-based algorithms, were ones of the first collaborative filtering algorithms developed [10]. Literature [11] uses a collaborative filtering recommendation method based on a deep latent factor model. This method uses deep matrix decomposition to solve the problem of recovering partially filled matrices in the collaborative filtering problem. However, it also has some shortcomings, such as the high cost of constructing the model. Literature [12] proposed a client/server framework to create a private recommendation system (PrivateRS). In the case that the ordinal

meaning of the rating is significantly blurred, the method can still generate accurate recommendations with acceptable losses. This method effectively utilizes the private mode of users or items, and can to a certain extent circumvent the privacy risks caused by the mining of user preferences by the recommendation algorithm. Literature [13] proposed a fuzzy clustering collaborative filtering method (FCCF) for time-aware POI recommendation to obtain higher POI performance. A fuzzy clustering based collaborative filtering algorithm (FCCF) is proposed for time-aware POI recommendation. The fuzzy c-means technique can reduce repeated calculation and comparison and is used to group similar users. The collaborative filtering technique also provides suggestions for a number of the top-N POIs at a specific time to a target user. The above methods can provide users with appropriate recommendation results when the amount of data sets is limited, but when the amount of data sets increases, they all face scalability problems. Algorithms based on neighbors do not need to build a specific model, but use a user score matrix to calculate the similarity between users or items, so the collaborative filtering algorithm based on neighbors is easier to implement. User score matrix is used to calculate the similarity between the users or items. The recommendation system combines the similarity determined by the score value with the similarity determined by the user score probability and the type of project to increase the accuracy of the similarity between users. The neighbor-based collaborative filtering algorithm first calculates the similarity between users (products) based on the user's historical information, and then uses the evaluation of other products by neighbors with higher similarity to the target user (product) to predict the user's preference for a specific product degree. The system recommends target users based on this degree of preference. Literature [14] proposed a UBCF method based on the coverage-based rough set theory. Compared with traditional UBCF, this method adds a user reduction process, which can remove redundant users among users. Literature [15] proposed a new method of similarity measurement method, based on the attributes of items to calculate the similarity between users. In order to calculate the similarity more accurately, the user's likes and dislikes of the similar attributes of a certain item are respectively considered. When there are no users with common ratings in the similarity data set, this method can have a good recommendation effect. Literature [16] proposed a user collaborative filtering method based on fuzzy C-means. In collaborative filtering, clustering technology can be used to group the most similar users into some clusters. Fuzzy clustering is one of the most commonly used clustering techniques. Compared with other clustering methods, it has a greater improvement effect on collaborative filtering methods. Combining the center of gravity defuzzification fuzzy clustering with the Pearson correlation coefficient improves the recommendation accuracy. Literature [17] proposes a method to find nearby users through subspace clustering. In this method, the author extracts different subspaces under the categories of interest, disinterest, disinterest, and disinterest. Based on the subspace, a tree structure of neighboring users is drawn for the target user. The problem of data sparseness in the system filtering algorithm based on nearest neighbors can be slightly alleviated. Literature [18] proposed a social recommendation

method based on adaptive neighbor selection mechanism on this basis. The user's initial neighborhood set is determined using this process, which combines historical ratings and social data about other users to build the user's initial neighborhood set. The initial rating of things that are invisible is predicted using these neighbor sets. A confidence model is also suggested in order to establish a new adaptive neighborhood set by removing pointless persons from the user's initial neighborhood. In order to forecast new invisible item ratings and suggest items of interest to active users, the new user-adaptive neighborhood set is used. The collaborative filtering method based on neighbors has been enhanced by the aforementioned techniques in a variety of ways, but there are still some drawbacks. For example, some models still cannot completely overcome the high dependence on user scores or the sparseness of the collaborative filtering method based on neighbors. It is difficult to find stable and reliable neighbors in the rating of user items, so the running time is long and the prediction accuracy drops sharply. The emergence of expert collaborative filtering algorithms largely compensates for this shortcoming.

3 METHODOLOGY

3.1 Problem Description

For the convenience of the following description, here is a unified description of the labels used in the text, as shown in Table1.

Symbol	Description
u, v	User u and user v
i, j	Item i and item j
I_u	A collection of items rated by user u
I_v	A collection of items rated by user v
U_i	A collection of users who rated item i
$\bar{r_u}$	The rating mean of user u
$\bar{r_v}$	The rating mean of user v
$r_{u,i}$	User u's rating for item i
$r_{v,i}$	Expert v's rating for item i
$P_{u,i}$	User u's prediction score for category c item i
$\bar{r_{u,c}}$	User u's average rating of category c items
$\bar{r_{v,c}}$	Average value of expert v's scores on category c
E_c	Experts in category c items

Table 1. Symbol and Description

Obtain the user rating matrix R in the historical purchase records, $R = r_{ui}$ $(1 \leq u \leq m, 1 \leq i \leq n)$. Where m represents the number of users and n represents the number of projects. If user A does not rate item i, then the value is 0, as shown

in Formula (1).

$$r_{u,i} = \begin{cases} r_{u,i}, & \text{if } \textit{rating}, \\ 0, & \text{if } \textit{no rating}. \end{cases} \tag{1}$$

Calculate the similarity between users in the training set to form a similarity matrix. Combined with the expert algorithm, according to the item i to be scored and the item category matrix, various signal processing methods are used to screen out the experts of the category of item i. Finally, the prediction score matrix R_{pred} is formed.

3.2 Model Introduction

Collaborative filtering (CF), as an important part of the recommendation system, has received extensive attention from the industry and academia [19]. The collaborative filtering algorithm uses the behavior of users similar to the user to be recommended to infer the user's preference for a specific product, and then makes corresponding recommendations based on this preference. At present, the collaborative filtering recommendation algorithm mainly includes two types: neighbor-based and model-based. Algorithms based on nearest neighbors directly use known scores to make predictions; while model-based algorithms learn prediction models from known scores [20].

3.2.1 Nearest Neighbor Algorithm

The nearest neighbor model has the advantages of simplicity, reasonability, efficiency and stability. The common framework for predicting items on a given user is based on the nearest neighbor method [21]. The basic principle of the nearest neighbor model is to find k nearest neighbors to replace the current user. In order to solve the problem of finding neighbors, we first need to find a way to express the relationship between users. The Pearson correlation algorithm is a memory-based collaborative filtering technique which solves the scalability issue by determining how similar two user-rated items are to one another or how similar two user-rated items are to one another. The distance relationship, PCC (Pearson Correlation Coefficient) is the most commonly used measurement algorithm for collaborative filtering algorithms based on neighbors, and its calculation formula is as Formula (2):

$$\text{sim}(u,v) = \frac{\sum_{i \in I_u \cap I_v} (r_{v,i} - \bar{r}_v)^2}{\sqrt{\sum_{i \in I_u \cap I_v} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_u \cap I_v} (r_{vi} - \bar{r}_v)^2}}. \tag{2}$$

3.2.2 Expert Algorithm

The expert algorithm, as the name suggests, divides items into categories, finds users in each category that have public reference significance for the recommendations of other target users, and defines them as experts. Expert users should meet the following definitions.

Definition 1. For project category A, expert E_c is defined by Formula (3):

$$|I_u| \leq |I_c| \left((u \in U - E_c, \forall v \in E_c\right). \tag{3}$$

The expert algorithm will satisfy the above-mentioned users to become expert users. The earliest expert algorithm consists of two parts: finding experts and generating recommended values. In actual operation, the recommendation effect of this structure is not ideal. As more and more professionals participate in research and improvement, the expert algorithm currently consists of three parts: finding experts, calculating similar values between experts and users, and generating recommended values.

Find the Experts. According to the item to be scored and the item-category matrix, determine the category of the item, calculate the number of times to evaluate all items of the category for all users, and arrange them in descending order. Descending order is a method of arranging integers from greatest to lowest. The first step in organizing the numbers is to start with the greatest number and work our way down to the smaller ones one by one. The number of experts is determined by the definition of experts and preset thresholds. Preset Threshold refers to the preset amount per Card that users have determined, up to which the Card is topped up each month and which the user may then use to consume food and beverages that month. The Preset Threshold will be supplemented each month by Available Funds.

Generate Recommended Value. In the expert algorithm of literature [14], when calculating the predictive score, only the expert suggestions with high similarity to the current user are considered, and the strategy of unconditionally trusting the expert-EA (expert algorithm) is adopted. When measuring the similarity between experts and users, Formula (4) is used, and the final score prediction uses the following Formula (4):

$$p_{u,i,c} = r_{u,c}^{-} + \frac{1}{k} \sum_{v \in E_c} \left(r_{v,i} - \bar{r}_{v,c}\right). \tag{4}$$

When the item to be predicted belongs to multiple categories, Formula (5) is used to calculate the score value.

$$p_{u,i,} = \frac{1}{|c_i|} \sum_{c \in c_i} p_{u,i,c}. \tag{5}$$

Among them: c is the number of categories to which the project belongs; P is the predicted value of each category. The running time of the above prediction scoring algorithm is relatively short, but it performs generally in terms of prediction accuracy. The difference between observed and predicted values should be used to calculate predictive accuracy. The projected values, however, may pertain to many types of knowledge. The consequent predictive accuracy can therefore be used to relate to many concepts. When the project is determined, the prediction score of this algorithm for different users is almost the same, because the expert's choice does not consider the current user, but only considers the items that the current user needs to predict.

Kurtosis Factor. Kurtosis k is a numerical statistic reflecting the distribution characteristics of random variables, a normalized 4^{th} order central moment, and a signal waveform characteristic. In mechanical principles, the kurtosis coefficient means that when fatigue failure occurs on the working surface of the bearing, the shock pulse generated at the defect of the working surface per revolution, the greater the failure, the greater the impact response amplitude, and the more obvious the failure phenomenon. Fatigue failures are closely correlated to components that withstand cyclic pressures or strains that permanently damage them. This builds up until a fracture forms, propagates, and leads to failure. Fatigue is the term used to describe the cyclic loading-induced process of damage development and failure. Shock Pulse Monitoring (SPM) is a patented predictive maintenance system that measures vibration and shock pulses of joints in motors to determine their condition and operational life before the next overhaul procedure. The kurtosis coefficient can represent the probability of the occurrence of large amplitude pulses caused by faults. The kurtosis coefficient is used to determine if a density is more or less peaked around its center than a normal curve, and negative values are frequently used to signify that a density is overstated around its center than a normal curve. This recommendation system introduces the kurtosis factor. Regarding the user's rating sequence as a random discrete signal, and then randomly sorting the user's ratings k times, the average kurtosis of the user is obtained, which can be written as Formula (6):

$$C_q = \frac{1}{k} \sum_{i=1}^{k} \frac{\frac{1}{N} \sum_{i=1}^{N} \left(|U_i| - \overline{(r_u)} \right)^4}{\overline{(R_{u\text{rms}}^4)}}. \tag{6}$$

After introducing the kurtosis factor, the method of finding experts is shown in Figure 1. The hollow blue circle on the left side of the figure represents the user, the orange hollow circle above represents the item, and the blue solid circle represents the user's rating of the item. The darker the color, the higher the score. It can be seen from the figure that the scores of different items are quite different, and they are considered experts.

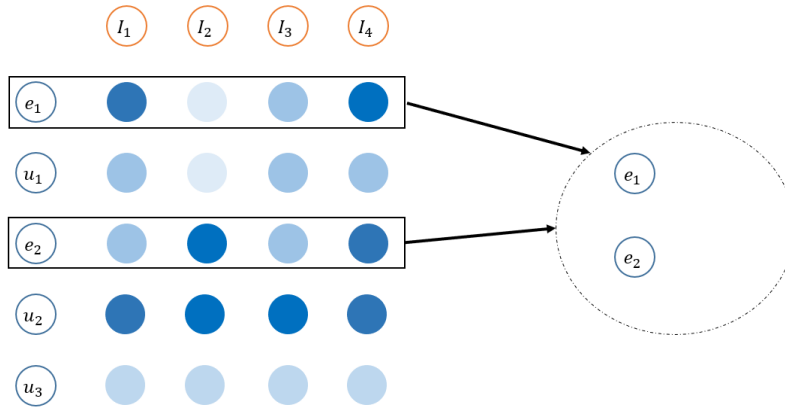


Figure 1. Schematic diagram of expert selection using kurtosis factor

4 EXPERIMENTS AND DATASETS

4.1 Datasets

MovieLens. This experimental data set uses the MovieLens data set provided by the GroupLens Research laboratory, which contains the ratings of movies by anonymous users, and each user has rated at least 20 of the movies. The rating value ranges from 1 to 5, with 1 indicating the lowest rating, 5 indicating the highest rating, and 0 indicating that the user has not rated the movie. In addition to scoring data, the data set also contains attributes of users and projects, such as the user's gender, age, occupation, project name, release year, style genre, etc. The style genre and scoring data of the user's movie are required for this experiment.

Jester. Jester was created by Ken Goldberg and his team at the University of California, Berkeley, and comprises about 6 million ratings for 150 jokes. Jester uses online user reviews to compile its ratings, just like MovieLens. Jester stands apart from other data sets in two ways: first, it has a continuous scale from −10 to 10, and second, it has the highest score density in terms of magnitude. What does "how many things are reviewed by each user" indicate in terms of the rating density? The rating density will be 100 % if each user has given a rating for every item. It will be 0 % if nothing has been rated. Jester has a 30 % density, which means that the majority of users only rated 30 % of the jokes. In example, MovieLens 1M has a density of 4.6 % (other data sets have densities of less than 1 %). Of course, it is not quite that easy. The quantity of goods that each user reviews varies. While most users only rate a small number of goods, other people rate numerous items.

Book-Crossings. Based on information from bookcrossing.com, Cai-Nicolas Ziegler developed the book score data set known as Book-Crossings. It has 1.1 million reviews from 90 000 individuals for 270 000 books. Cai-Nicolas Ziegler collected the BookCrossing (BX) dataset during a 4-week crawl (August/September 2004) with permission of Ron Hornbaker, CTO of Humankind Systems. It has 1 149 780 ratings (explicit or implicit) of 271 379 books from 278 858 individuals who have been anonymized but have demographic information. User reviews of books are gathered in the BookCrossing dataset. It has both explicit ratings (from 1 to 10 stars) and implicit ratings (based on how readers interacted with the book). The score, which includes explicit and implicit scores, is between 1 and 10. One of the least dense data sets, and the least dense data set with a score, is the Book-Crossings data collection.

Last.fm. A collection of music recommendations is provided by Last.fm. A list of the top artists and the quantity of plays are provided for each user in the data set. A tag designed to incorporate all music available on last.fm, regardless of category, and to promote a sense of community. When finished, the "all" tag should give access to everyone an incredibly diverse range of music. Additionally, it has tags that users have added that can be conducted to generate content vectors. Some information (about a particular song or the time someone is listening to the music) will be lost after the Last.fm data has been aggregated. In these examples, it is the only data set with information about the user's social network.

4.2 Statistics

We have conducted statistics on the four data sets to better understand the datasets. Statistics, also known as the "Science of Facts", allows us to generate conclusions from a set of data. Additionally, it may help people across all sectors in obtaining responses to their research or business-related queries and predict results, such as what program they might want to watch on their preferred video app next. Figure 2 shows a histogram of the number of users and the number of items in the four datasets. For the convenience of observation, we use thousands as the unit, and only part of the histogram of the Book-Crossings dataset is shown. As can be seen from the figure, the number of users of the Jester dataset is far greater than the number of items. The number of users in the Last.fm dataset is much smaller than the number of items. The number of users and items of MovieLens and Book-Crossings is relatively balanced. In addition, we can also see that the number of samples in the Book-Crossings dataset is much larger than the other three datasets.

4.3 Evaluation Metric

The criteria for evaluating the prediction accuracy of the recommendation system are divided into two categories: decision-making accuracy standards and statistical

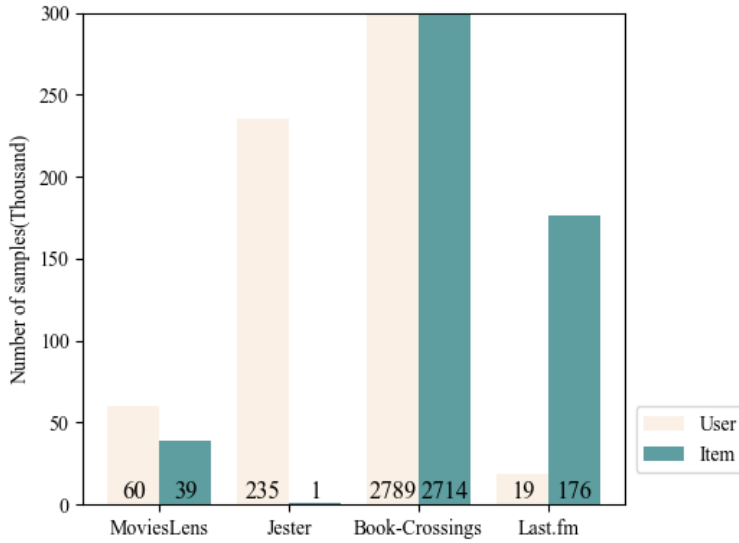


Figure 2. Number of users and items in the four datasets

accuracy standards. The actual response of an organization to a task is compared to the required response, and accuracy is calculated by allocating a cost to the difference. The organization's capacity to respond within a time frame established between the demands of the work is reflected in its ability to be responsive. A series of measurements are evaluated for accuracy to determine if they are generally accurate. This paper adopts the root mean square error (RMSE), which is sensitive to the response of very large or very small errors. The root mean square error or root mean square deviation is one of the most often employed metrics for assessing the accuracy of predictions. It illustrates the Euclidean distance between measured true and predicted values. In recommendation systems, RMSE is widely used as a common measurement error standard. The principle is to calculate the square root of the ratio of the user's projected value and the true value of the project to the square root of the ratio of the number of users n, as shown in Formula (7):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{m}(h(x_i) - x_i)^2}. \tag{7}$$

4.4 Experiments Environment

We use Python 3.6 as the programming language for method implementation. We use Pytorch 1.7.1 to implement the neural network. We use Scikit-learn 0.24.0 to implement machine learning. Scikit-learn is a free machine learning package for

Python. It supports a number of techniques, including support vector machines, random forests, and k-neighbors, as well as the NumPy and SciPy libraries from Python. The most efficient and dependable machine learning library is Python's Scikit-learn (Sklearn) package. Through a Python interface, it provides a range of efficient techniques for statistical modeling and machine learning, including dimensionality reduction, clustering, and regression. All pre-trained models are loaded from Hugging Face Transformers. Hugging Face Transformers is a platform that offers the community APIs to access and use cutting-edge pre-trained models accessible through the Hugging Face hub. PreTrained Model is responsible for maintaining the configuration of the models and handles methods for loading, downloading, and saving models as well as a few methods that are common to all models. These methods can be used to load or save a model from a local file or directory or from a pretrained model configuration that the library distributes. The GPU used for model training is GTX1660 6G.

5 EXPERIMENTAL RESULTS AND ANALYSIS

We conduct experiments on the above five datasets to verify the advanced nature of our method. The KNN algorithm does not work well with large datasets. The cost of computing the distance between the new point and each existing point is prohibitively expensive and it also reduces the performance. The KNN technique must be used to any dataset after feature scaling (standardization and normalization). It can be seen from Table 2 that our method surpasses the KNN method on all four data sets. Among them, our method obtains the best result on the Jester dataset, that is, the root mean square error is 0.15. And our method has obtained the most effect improvement on the Last.fm data set, that is, the mean square error is reduced by 0.06. In addition, our method improves by 0.05 on the MovieLens dataset, 0.03 on the Jester dataset, 0.05 on the Book-Crossings dataset, and 0.06 on the Last.fm dataset. This fully demonstrates that our method has universal applicability on a variety of data sets.

Dataset	Method		Promote
	KNN	Ours	
MovieLens	0.23	0.18	−0.05
Jester	0.18	0.15	−0.03
BookCrossings	0.25	0.20	−0.05
Last.fm	0.22	0.16	−0.06

Table 2. Dataset, Method, and Promote

We try different k to get the best kurtosis coefficient. Figure 3 shows the experimental results obtained by using different kurtosis coefficients on four datasets. Kurtosis is observed in a symmetric distribution, and its predicted value is 3. Positive kurtosis is indicated by a kurtosis value larger than three. The range of the

kurtosis value in this situation ranges from 1 to infinity. It can be seen that our method can obtain the best effect when the kurtosis coefficient is about 5.

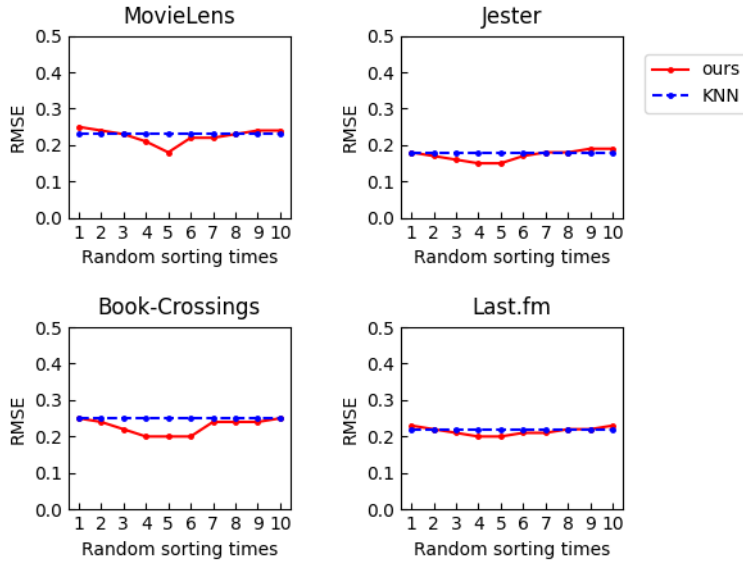


Figure 3. The effect of k on the results

6 TIME COMPLEXITY ANALYSIS

Although the method proposed in this paper can effectively improve the recommendation effect. However, the introduction of additional calculations will also increase the time complexity. In this article, the influencing factor most closely related to time complexity is the length of the user rating sequence. The term "temporal complexity" refers to how many operations an algorithm uses to complete a task (considering that each operation takes the same amount of time). The algorithm that completes the job with the fewest operations is thought to be the most effective one in terms of time complexity. Figure 4 shows the changes in the training time and test accuracy of the model as the length of the user rating sequence increases. Among them, the abscissa represents the length of the user rating sequence. The ordinate on the left represents the time required for the model to complete the entire training process. The ordinate on the right represents the accuracy of the model on the test set. The GPU used for training here is GTX1660 6G.

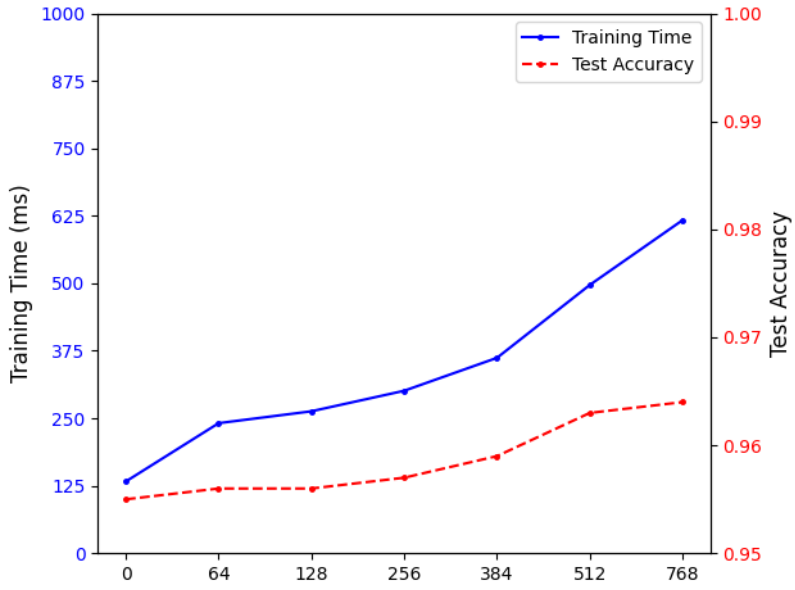


Figure 4. Time complexity graph

7 CONCLUSIONS AND FUTURE

This paper proposes a recommendation algorithm that combines expert algorithms and collaborative filtering algorithms. This method uses expert algorithms to screen out more valuable users as expert users, and uses the numerical statistics that can reflect the distribution characteristics of random variables in the mechanical field. The degree factor is used as a criterion to measure whether an expert is qualified. Collaborative filtering algorithm, as an evergreen algorithm in the field of recommendation systems, can meet the individual needs of users to a large extent, and is complementary to expert algorithms. Experiments on multiple data sets of MovieLens, Jester, Booking Crossings, Last.fm show that this method can effectively improve the recommendation accuracy and is reliable.

In the future, we will continue to study the sparseness, interpretability, and relational reasoning of recommendation systems, and devote ourselves to designing models that take into account indicators such as popularity, diversity, operational strategies, and logic. Specifically, the current algorithm we design is based on collaborative filtering, but the method proposed in this paper does not make full use of the advantages of model-based methods. Model-based methods have strong advantages in the face of data sparsity. Therefore, this article intends to incorporate the idea of using kurtosis factors to screen experts into both the nearest neighbor-based

and model-based methods. We believe that this paper may effectively improve the accuracy of recommendation in the case of insufficient data.

REFERENCES

[1] Hong, H.—Kim, H. J.: Antecedents and Consequences of Information Overload in the COVID-19 Pandemic. International Journal of Environmental Research and Public Health, Vol. 17, 2020, No. 24, Art. No. 9305, doi: 10.3390/ijerph17249305.

[2] Alhijawi, B.—Al-Naymat, G.—Obeid, N.—Awajan, A.: Novel Predictive Model to Improve the Accuracy of Collaborative Filtering Recommender Systems. Information Systems, Vol. 96, 2021, Art. No. 101670, doi: 10.1016/j.is.2020.101670.

[3] Su, X.—Khoshgoftaar, T. M.: A Survey of Collaborative Filtering Techniques. Advances in Artificial Intelligence, Vol. 2009, 2009, Art. No. 421425, doi: 10.1155/2009/421425.

[4] Li, Z.—Zhang, L.: Fast Neighbor User Searching for Neighborhood-Based Collaborative Filtering with Hybrid User Similarity Measures. Soft Computing, Vol. 25, 2021, No. 7, pp. 5323–5338, doi: 10.1007/s00500-020-05531-1.

[5] Huang, X.—Wang, G.: Learning Recommendation Based on Hybrid Collaborative Filtering Algorithm. Journal of Physics: Conference Series, Vol. 1629, 2020, No. 1, Art. No. 012008, doi: 10.1088/1742-6596/1629/1/012008.

[6] Wang, H.—Wu, J.: Collaborative Filtering-Based Film Recommendation Technique Utilizing Time and Film Genres. Journal of Physics: Conference Series, Vol. 1631, 2020, No. 1, Art. No. 012105, doi: 10.1088/1742-6596/1631/1/012105.

[7] Amatriain, X.—Lathia, N.—Pujol, J. M.—Kwak, H.—Oliver, N.: The Wisdom of the Few: A Collaborative Filtering Approach Based on Expert Opinions from the Web. Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '09), 2009, pp. 532–539, doi: 10.1145/1571941.1572033.

[8] Hwang, W. S.—Lee, H. J.—Kim, S. W.—Lee, M.: On Using Category Experts for Improving the Performance and Accuracy in Recommender Systems. Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM '12), 2012, pp. 2355–2358, doi: 10.1145/2396761.2398639.

[9] Cho, J.—Kwon, K.—Park, Y.: Collaborative Filtering Using Dual Information Sources. IEEE Intelligent Systems, Vol. 22, 2007, No. 3, pp. 30–38, doi: 10.1109/MIS.2007.48.

[10] Yan, Y.—Xie, H.: Collaborative Filtering Recommendation Algorithm Based on User Preferences. Journal of Physics: Conference Series, Vol. 1549, 2020, No. 3, Art. No. 032147, doi: 10.1088/1742-6596/1549/3/032147.

[11] Hwang, W. S.—Lee, H. J.—Kim, S. W.—Won, Y.—Lee, M. S.: Efficient Recommendation Methods Using Category Experts for a Large Dataset. Information Fusion, Vol. 28, 2016, pp. 75–82, doi: 10.1016/j.inffus.2015.07.005.

[12] Liu, Q.—Cheng, B.—Xu, C.: Collaborative Filtering Based on Star Users. 2011 IEEE 23rd International Conference on Tools with Artificial Intelligence, 2011, pp. 223–228, doi: 10.1109/ICTAI.2011.41.

[13] Pham, X. H.—Jung, J. J.—Nguyen, N. T.: Integrating Multiple Experts for Correction Process in Interactive Recommendation Systems. In: Nguyen, N. T., Hoang, K., Jędrzejowicz, P. (Eds.): Computational Collective Intelligence. Technologies and Applications (ICCCI 2012). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 7653, 2012, pp. 31–40, doi: 10.1007/978-3-642-34630-9_4.

[14] Pham, X. H.—Nguyen, T. T.—Jung, J. J.—Nguyen, N. T.: <A,V>-Spear: A New Method for Expert Based Recommendation Systems. Cybernetics and Systems, Vol. 45, 2014, No. 2, pp. 165–179, doi: 10.1080/01969722.2014.874822.

[15] Gogna, A.—Majumdar, A.: Matrix Completion Incorporating Auxiliary Information for Recommender System Design. Expert Systems with Applications, Vol. 42, 2015, No. 14, pp. 5789–5799, doi: 10.1016/j.eswa.2015.04.012.

[16] Li, Y.—Wu, H.—Li, L.—Yang, Y.—Zhang, C.—Bie, R.: Improved Field of Experts Model for Image Completion. Optik, Vol. 142, 2017, pp. 174–182, doi: 10.1016/j.ijleo.2017.05.077.

[17] Nikzad-Khasmakhi, N.—Balafar, M. A.—Feizi-Derakhshi, M. R.: The State-of-the-Art in Expert Recommendation Systems. Engineering Applications of Artificial Intelligence, Vol. 82, 2019, pp. 126–147, doi: 10.1016/j.engappai.2019.03.020.

[18] Pujahari, A.—Sisodia, D. S.: Aggregation of Preference Relations to Enhance the Ranking Quality of Collaborative Filtering Based Group Recommender System. Expert Systems with Applications, Vol. 156, 2020, Art. No. 113476, doi: 10.1016/j.eswa.2020.113476.

[19] Ortega, F.—González-Prieto, Á.: Recommender Systems and Collaborative Filtering. Applied Sciences, Vol. 10, 2020, No. 20, Art. No. 7050, doi: 10.3390/app10207050.

[20] Zhang, Z.—Zhang, Y.—Ren, Y.: Employing Neighborhood Reduction for Alleviating Sparsity and Cold Start Problems in User-Based Collaborative Filtering. Information Retrieval Journal, Vol. 23, 2020, No. 4, pp. 449–472, doi: 10.1007/s10791-020-09378-w.

[21] Srifi, M.—Oussous, A.—Ait Lahcen, A.—Mouline, S.: Recommender Systems Based on Collaborative Filtering Using Review Texts – A Survey. Information, Vol. 11, 2020, No. 6, Art. No. 317, doi: 10.3390/info11060317.

Shuo Wang is currently working toward her Ph.D. degree in computer science and technology at Harbin Engineering University, China. She majors in recommendation system and deep learning.

Jing Yang is Professor, Doctoral Supervisor in the College of Computer Science and Technology at the Harbin Engineering University, China. She obtained her Ph.D. from the College of Computer Science and Technology at Harbin Engineering University, China. Her research interests include database theories, data mining, data stream and privacy protection. She has published over 130 papers in journals and conferences.

Fanshu Shang is currently working toward her Ph.D. degree in computer science and technology at the Harbin Engineering University in China. Her research areas mainly include deep learning and privacy.

Jingyun Sun is a doctoral student majoring in computer science at the Harbin Engineering University, majoring in natural language processing.

Computing and Informatics, Vol. 42, 2023, 878–895, doi: 10.31577/cai_2023_4_878

ARCHITECTURE OF A FUNCTION-AS-A-SERVICE APPLICATION

Ondrej Habala*, Martin Bobák, Martin Šeleng,
Viet Tran, Ladislav Hluchý

Institute of Informatics
Slovak Academy of Sciences
Dúbravská cesta 9
845 07 Bratislava, Slovakia
e-mail: {ondrej.habala, martin.bobak, martin.seleng, viet.tran, ladislav.hluchy}@savba.sk

Abstract. Serverless computing and Function-as-a-Service (FaaS) are programming paradigms that have many advantages for modern, distributed and highly modular applications. However, the process of transforming a legacy, monolithic application into a set of functions suitable for a FaaS environment can be a complex task. It may be questionable whether the obvious advantages received from such a transformation outweigh the effort and resources spent on it. In this paper we present our continuing research aimed at the transformation of legacy applications into the FaaS paradigm. Our test subject is an airport visibility system, a sub-class of the meteorological services required for airport operations. We have chosen to modularize the application, divide it into parts that can be implemented as functions in the FaaS paradigm, and provide it with a simple cloud-based data management layer. The tools that we are using are Apache OpenWhisk for FaaS and Airflow for workflow management, Apache Airflow for workflow management and NextCloud for cloud storage. Only a part of the original application has been transformed, but it already allows us to draw some conclusions and especially start forming a generalized picture of a Function-as-a-Service application.

Keywords: FaaS, serverless computing, cloud computing

Mathematics Subject Classification 2010: 68-U35

* Corresponding author

1 INTRODUCTION

In this paper we present the initial stages of the construction of an airport visibility meteorological application based on the Function-as-a-Service paradigm [1]. The application has been in use by its developer and operator (MicroStep-MIS) for several years now at multiple airports, albeit as a monolithic system, without the use of cloud or serverless computing. Now it is being attempted to transform it into a serverless application. It is not necessary for the application's work, but it is seen by the developer as an investment into the commercial future of the product, as it will:

- allow the developer to modularize the application, potentially offering several different deployments with different requirements, functionality and pricing,
- allow to deploy the application without requiring the customer to invest into hardware and maintenance,
- allow to develop the application into a completely different functionality and even domains,
- give the developer critical know-how based on modern computing paradigms, which will allow it to stay current with its products, as more and more customers move from dedicated hardware to cloud computing and even serverless computing.

The re-development of the application in the FaaS paradigm is done in cooperation with the Institute of Informatics of the Slovak Academy of Sciences – the research partner. For the research partner the development is of interest because of its long-standing research in distributed computing, leading to cloud computing and now serverless computing. While the application's parameters may not be ideal for the serverless paradigm, it is still very well usable, and will benefit from the application of FaaS concepts.

The development of the serverless version of the application is still in its initial, exploratory stages, and the contents of this article reflects this fact. In the following chapters we present the general overview of what is Function-as-a-Service, as well as of the tools we are currently using – OpenWhisk, OSCAR, OpenFaaS and Airflow. We also describe the general architecture and workings of the application as well as the challenges it faces during the process of moving towards a serverless architecture based on FaaS. In the final chapter of the paper we present our future plans for the application and its further transformation.

2 THE FUNCTION AS A SERVICE PARADIGM

Function as a service (FaaS) is a subset of serverless computing [2] that provides a platform allowing developers to write and deploy applications without building and maintaining the underlying infrastructure. Typical tasks related to infrastructure management like resource provisioning, maintenance and regular update of base

operating systems are on the responsible Cloud provider. Developers may focus only on application codes and their logics.

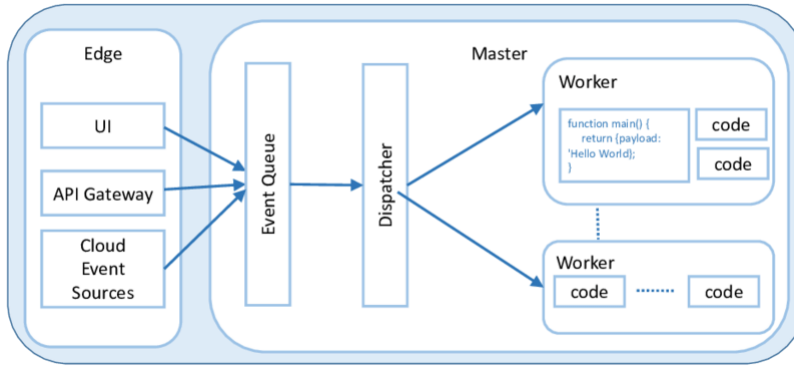


Figure 1. Serverless platform architecture [2]

The architecture of FaaS is shown in Figure 1. The core component of the FaaS platform is the event dispatcher that receives events from different sources: users' requests sent to API gateways, mouse clicks on GUI, time alarms, data arrivals on cloud storages and so on. For each event, application developers can define the corresponding code that will be executed by the dispatcher on worker nodes when the event occurs. An event queue is often placed in front of the event dispatcher for handling high event rates, when all worker nodes are busy in processing other events.

Whole system, including the dispatcher and worker nodes are managed by the FaaS provider so application developers may focus only on the codes that are called by the event dispatcher when an event occurs. The codes corresponding to the events are typically implemented as functions: stateless small pieces of code, therefore the platform is called "Function as a Service". As the code is stateless, dispatchers can execute multiple instances of the code in parallel with very low overheads, so the application is scaled easily, and practically limited only by the computation capacity of the FaaS providers. The billing system is based on the actual resource usage by the functions, if there is no function call, no cost occurs.

Functions have to be implemented in a programming language supported by the FaaS platforms and using libraries provided by the providers. That limits portability of the code and makes potential vendor lock-in. Some FaaS platforms, e.g. OpenWhisk, can support Docker containers as functions so developers can implement the functions in any programming language and with custom libraries. Loading Docker containers at each execution causes much higher costs than function calls, so the containers are often cached and used repeatedly. The FaaS paradigm is also well suited for low-code [3] programming, as the serverless concept frees the developer of a whole class of details (code and data location, connections, etc.).

In summary, the main advantages of FaaS:

Automatic scaling: With FaaS, functions are scaled automatically, independently, and instantaneously according to the actual demands. That will relieve developers from concerns of high traffic or heavy use. The cloud providers will handle all issues related to scaling, e.g. allocating computing resources, replicating the codes and so on.

Cost efficiency: Users have to pay only for the computing resources they really use, not for idle resources that are often reserved for handling possible high demands in the typical IaaS (Infrastructure as a Service). As mentioned above, functions are scaled up automatically on high demands and scaled back down on low demands. If there are no demands or events, no costs are incurred.

Quick development: As developers do not need to manage infrastructure, they can focus only on the code, reducing the cost of development and the time to market.

FaaS also have some disadvantages:

Potential vendor lock-in: The application codes are built on the top of a concrete FaaS platform and difficult to port to another vendor.

Difficulties for testing: The codes are running on the top of a FaaS platform, it may make difficulties for creating local test environments for applications.

FaaS is very suitable for applications that have dynamic or volatile loads as it can scale easily and handle very high demands without big issues, and also has no cost when the application is idle. For applications with constant loads, the cost of FaaS may be higher than typical IaaS solutions.

The first commercial provider offering FaaS is Amazon AWS with AWS Lambda platform [4], followed by Google with Google Cloud Functions [5], Microsoft with Azure Functions [6]. In this paper, we will focus on the open-source FaaS platform OpenWhisk from IBM (described in Section 4).

3 THE APPLICATION – AIRPORT VISIBILITY USING CAMERAS

An important element in the safety of all kinds of transport is good visibility. Poor visibility can cause fatal accidents [7]. Visibility measurement is therefore a relevant issue for air transport during the whole flight but especially when the aircraft is maneuvering on or near the ground [8]. Aircraft accidents due to bad weather conditions comprise almost 50 % of all cases and the main cause of weather-related accidents is reduced visibility [9]. Good visibility information can significantly decrease the risk of accidents, number of redirected flights, save fuel and decrease negative economic consequences.

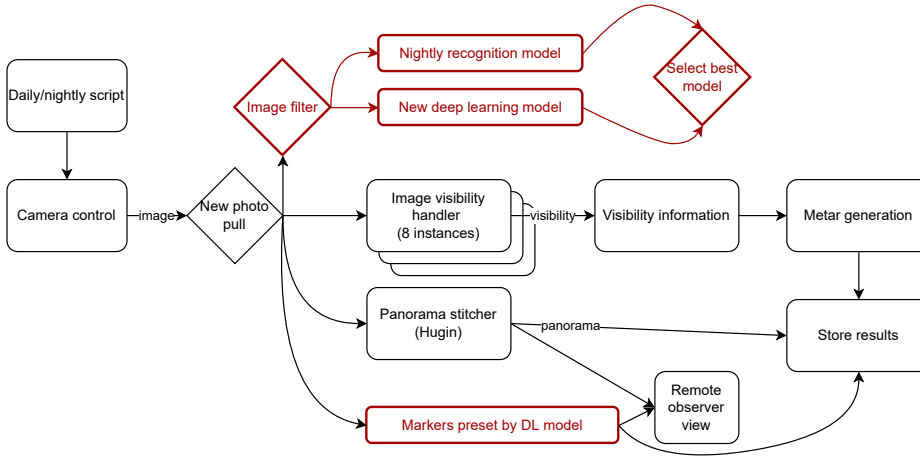


Figure 2. Architecture of the airport visibility application

There exist dedicated sensors for measuring visibility at airports [10], but:

1. they are usually costly to obtain,
2. their coverage is often insufficient,
3. they can only measure visibility at an exact measuring point.

Another approach to determine visibility is based on the research of remote and automatic visibility observation using camera images [11, 12]. This approach addresses all above mentioned problems, as:

1. the required cameras are cheaper than dedicated sensors,
2. cameras can cover larger environment,
3. cameras are often already present at airports and can be re-tasked for visibility measurements.

Visibility measurement using cameras includes instant processing of large number of images from various sources, with image recognition and automatization of modeling processes as well as the application of multiple parallel functionalities and checking mechanisms. Therefore a more sophisticated and highly flexible information infrastructure is necessary.

In our system for visibility determination, the camera images are the basis for a remote observer to estimate visibility with the help of reference points. In the automatic version images taken during good visibility are used to construct reference objects. Then during measurement the system determines which reference objects are visible and which are not, and from that information it can make the decision on visibility conditions. This system can monitor visibility according to aviation

rules for human observers, which is preferable to other arbitrary solutions. Since the responsibility of the system is depending on the capacity of the underlying infrastructure, it can be increased by expanding this infrastructure. It is possible to achieve frequency of visibility measurements of one minute or even less for critical situations – and by altering the frequency requirements, resources and costs can be saved when high frequency is not necessary.

The processes of our visibility measurement application are depicted in Figure 2. Camera configurations for image capture during a day and during a night differ, as there are differences in visibility reporting process by meteorological observers during day and night. Overall the cameras mimic manual observations. Images covering full horizon are taken and processed by the Image visibility handler module. This module finds visible reference points by comparison with the database of available reference points for every image, using edge detection as poor visibility presents itself with loss of contrast and thus loss of edges in the obtained images. Then the Visibility information module determines the minimal and prevailing visibility using recognized reference points. Prevailing visibility is the maximum distance visible throughout at least half of the horizon. Then information about prevailing visibility is displayed as in the METAR[1] reporting tool and results are stored in a database. In parallel runs the process of stitching panorama for remote observers to have better conditions for visibility observations remotely is run and outputs from these observations are also fitted in a METAR report and stored in the results database. We also plan to create new deep learning visibility model that could run in parallel to Image visibility handler module so we can evaluate this method.

The technologies used in the original application have to be modified to fit Function-as-a-Service infrastructure in order to be executed using OpenWhisk or OSCAR frameworks. Image data are acquired using the ONVIF standard. The user interface is provided through a web application server integrated within the IMS [13] software. The UI for IMS is built using industry proven standards: HTTP/HTTPS protocol, HTML and XML formats, JavaScript and AJAX technologies, which makes them well suitable for cloud deployment. The backend implementation uses Java. The UI, since it is HTTP based, is ready for access over the network. The IMS system is also accessible through a web browser. The basic IMS server software also allows for both edge and cloud deployment.

4 APACHE OPENWHISK

OpenWhisk is a free and open implementation of the Function-as-a-Service paradigm described in Section 2. The history of OpenWhisk is tied to the Amazon Web Services' Lambda service, first presented in November 2014. According to Rodric Rabbah, then working at IBM Research, his research group quickly realized the

[1] Standardized message for reporting meteorological conditions obligatorily emitted every half an hour by airports and meteorological institutions.

Lambda's value and the overall value of the serverless computing premise. He saw it as a transformation of computing native to cloud, and the future of the architectures of cloud applications [14]. Therefore, IBM Research started working on a competing product called Whisk. Whisk was later renamed OpenWhisk, open-sourced and transferred to the Apache Software Foundation Incubator [15]. Nowadays it is an open-source alternative to AWS Lambda [4], Microsoft Azure Functions [6] and Google Cloud Functions [5].

4.1 The OpenWhisk Programming Model

OpenWhisk is an event-driven system, in which events from an event source feed into triggers, and via rules cause the execution of actions (functions). The production of event can be done via a command-line tool, via a HTTP call to the OpenWhisk installation, or from numerous existing plugins for various tools, messaging services, and software systems that produce events. One OpenWhisk installation can serve several independent systems of actions, rules and triggers, each in its own namespace. The schema of this process is shown in Figure 3.

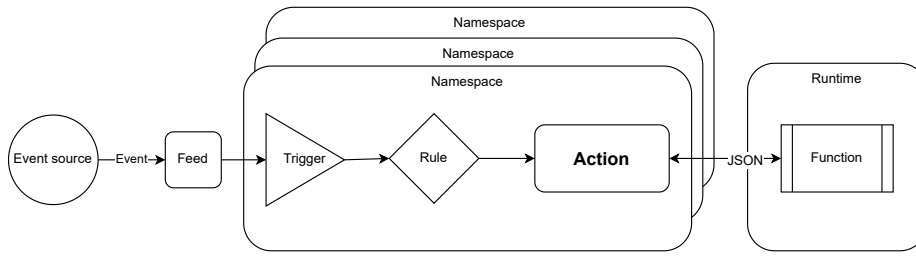


Figure 3. The event-driven programming model of OpenWhisk

The action itself (the principal function that is to be executed as a result of an event occurring) is provided by the developer of the system which uses OpenWhisk. It is a self-contained section of programming code, written in one of several supported programming languages:

1. Java,
2. Javascript,
3. Python,
4. PHP,
5. Go,
6. .NET,
7. Ruby,
8. Swift.

These languages are supported by their own runtime environments, implemented as docker images. Apart from support for functions implemented in these languages, a developer can also provide an action designed as a black box, via providing a specifically constructed docker image based on a provided Docker Runtime[2] image.

As we can infer from this description, each action is instantiated and executed in its own docker container, which contains the necessary software to compile and execute code in the programming language in which the action is written. In the case of a black-box docker action, the docker image itself contains the function, and the OpenWhisk execution system does not provide support with its compilation, it only calls the code implementing the action via an established mechanism.

Independently of which language (and, consequently, which runtime) is used, the process of execution of the action's code is similar:

- The runtime (docker image) contains a REST-enabled server.
- This server receives a JSON[3] structure with input data and the code of the action itself.
- The code of the action is compiled (if necessary) and executed, with the received input data provided to it.
- If the code of the action finishes in certain pre-set time, its result is packaged in JSON, and returned as the result of the initial call to the REST server.
- If the code of the action takes more time, the REST server returns a result indicating that the action is still executing. In that case, the result of the action can be queried later via the action's ID.

In the case of the black box action based on Docker Runtime, no compilation of code is done in the 3rd step and the input data is fed directly to an executable place already present in the docker image.

4.2 Installation and Use of OpenWhisk

OpenWhisk can be installed in several ways, as detailed in OpenWhisk documentation[4]. We have chosen to use Kubernetes installation. OpenWhisk is already available as a Helm chart, so deployment in Kubeapps is very straightforward. Additionally a local installation of the wsk command-line tool is required on the machine, from which the OpenWhisk installation is to be used. The installation and configuration of this tool is also described in OpenWhisk documentation[5].

[2] Apache OpenWhisk runtimes for docker, `https://github.com/apache/openwhisk-runtime-docker#readme`

[3] What is JSON, `https://www.w3schools.com/whatis/whatis_json.asp`

[4] OpenWhisk documentation: deployment options, `https://openwhisk.apache.org/documentation.html#openwhisk_deployment`

[5] OpenWhisk documentation: OpenWhisk CLI (wsk), `https://openwhisk.apache.org/documentation.html#wsk-cli`

4.3 Using OpenWhisk for Airport Meteorology

In the course of our work, we will implement all the modules of the application as shown in Figure 2 as OpenWhisk actions, to be managed and executed either programatically, or via a workflow orchestration system (see Section 5 for one such system which we intend to use).

So far, we have implemented 3 functions:

1. ImageVisibilityHander, as a Java action,
2. Visibility info from 8 xmls, also as a Java action,
3. Panorama stitch, as a black-box docker action, since we use a third party software (Hugin[6]).

In the case of the Panorama stitch action, we have created a specific docker image, based on the above described Docker Runtime image. Our image contains also the Hugin software and a program, which:

1. receives the JSON input structure from the Docker Runtime proxy via the HTTP protocol,
2. decodes it into a set of parameters, including input file URLs,
3. downloads the input files,
4. executes panorama stitching on the downloaded input files using the Hugin software's capabilities,
5. uploads the resulting stitched panorama to cloud storage,
6. indicates success or failure of the panorama stitching operation on output.

The details of the modification of the original Docker Runtime from OpenWhisk are detailed by J. Thomas in [16].

In the case of the two Java-based actions from the above list, we also had to perform additional steps. The software that is necessary to execute the actions requires several third party libraries, including OpenCV[7], which are quite large. The maximum size of an action in OpenWhisk is 48 MB, and this is not sufficient to transfer the software and all the required libraries. Therefore, we had to create our own specific Java Runtime for OpenWhisk actions, based on the standard OpenWhisk Java Runtime[8] provided by the OpenWhisk project. The building of the docker image of all OpenWhisk runtimes is managed by the Gradle[9] build tool. The libraries can be added from an external repository, as shown in [17], or as local files, according to the rules of Gradle build files. In our case we have used local JAR files

[6] Hugin – Panorama Photo Stitcher, `https://hugin.sourceforge.io/`

[7] The Open Source Computer Vision Library, `https://opencv.org/`

[8] OpenWhisk Java Runtime, `https://github.com/apache/openwhisk-runtime-java#readme`

[9] Gradle build tool, `https://gradle.org/`

provided by the developers of the application. The resulting source code for our application-specific Java Runtime is available in our GitHub repository[10].

5 SERVERLESS FUNCTIONS ORCHESTRATION WITH APACHE AIRFLOW

Many scientific applications are so complex that they are often described by complex workflows requiring efficient management and coordination of jobs. They process and analyze large volumes of data through numerous interconnected tasks. The management and organization of the tasks are challenging due to their complexity and their need for scalability and reliability.

Function as a Service (see Section 2) changes the development of applications, allowing developers to focus on code instead of infrastructure management. Apache AirFlow [18] is an open source[11] framework for the management of lightweight serverless functions. It amalgamates individual tasks into a workflow that is expressed as a directed acyclic graph (DAG) representing the workflow structure and dependencies between tasks. Each task within the DAG can be associated with a serverless function, enabling the integration of the FaaS paradigm into workflows.

5.1 AirFlow Architecture

Apache Airflow is composed of the following components (see Figure 4):

Scheduler: orchestrates the execution of tasks defined by a DAG. Execution order is defined by their dependencies, and tasks are triggered according to them.

Workers: execute tasks within a specific environment (e.g. local machine, cluster, or cloud).

Web server: provides a user interface for AirFlow. Users can view and manage workflows, monitor task execution, logs, and metadata of workflow runs.

Metadata database: stores and manages information related to DAGs, tasks, workflows and their executions. It maintains the state and history of workflows.

5.2 Using AirFlow for Airport Meteorology

In our case, the tasks are serverless functions, since the requirements for AirFlow tasks are atomicity and no resource sharing, which the serverless functions meet. The resulting workflow characterizes the relations between its tasks and also defines their execution order.

The paper presents the Airflow workflow (see Figure 5) for the airport meteorology application (see Section 3). The whole application is divided into individual

[10] `https://github.com/IISAS/openwhisk-runtime-java`

[11] Under Apache-2.0 license.

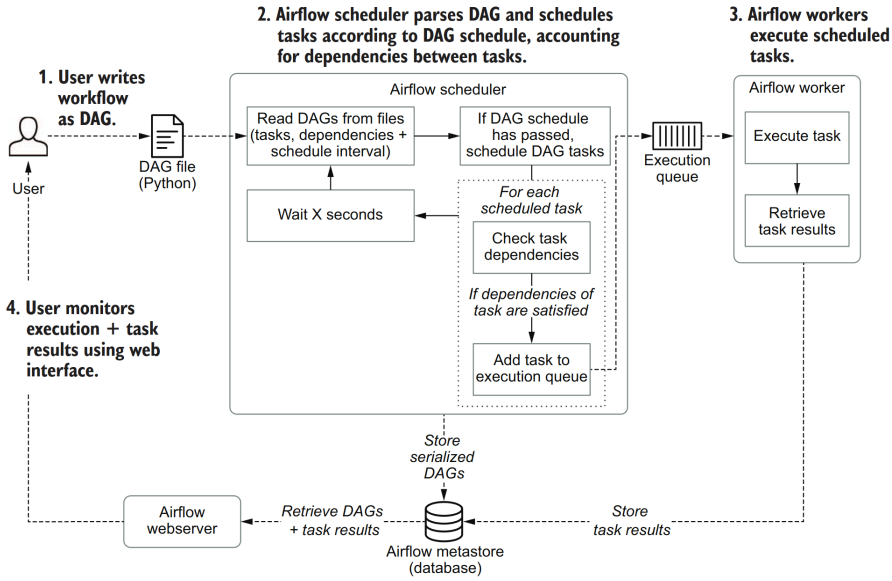


Figure 4. Schematic overview of the process involved in developing and executing pipelines as DAGs using Airflow [19]

tasks which are implemented as Docker containers. The workflow is divided into the following tasks:

1. **day_or_night**: determines whether the captured images fall within the day or night period. This task plays a crucial role in optimizing operational procedures and decision making in aviation workflows.
2. **image_from_camera**: retrieves airport visibility data from selected airport camera data sources. The task integrates real-time visual data into the workflow, allowing further analysis, processing, or decision-making based on the captured images. The images are sent to the following tasks: DL_model_markers, Image_filter_1, Panorama_stitch, and Image_Visibility_Handler.
3. **DL_model_markers**: applies deep learning (DL) models to detect and analyze markers or specific features within the images.
4. **Image_filter_1**: applies a specific image filter (in this case filter #1) to enhance or modify the images from the camera. This task enables the integration of image processing algorithms to manipulate them.
5. **Panorama_stitch**: is responsible for stitching multiple overlapping images together to create a seamless panoramic image. This task uses image processing algorithms and computer vision techniques to align and blend the input images, producing a panoramic view of the airport.

6. **Image_Visibility_Handler**: analyzes and assesses the visibility conditions in images covering the full 360° horizon. This task takes advantage of image processing techniques and visibility assessment algorithms to determine the level of visibility or clarity in the input images.

7. **Visibility_informations**: gathers and provides relevant visibility-related information to observers. This task collects data, generates insights, and presents visibility metrics or indicators derived from the camera images.

8. **Metar_generation**: produces a METAR (Meteorological Aerodrome Report) data based on relevant meteorological parameters. This task gathers weather information, processes it, and generates a standardized METAR report that provides essential weather observations for an airport.

9. **Remote_Observer_View**: enables remote monitoring and observation of an airport area using visual data captured from remote cameras. This task facilitates real-time or near-real-time viewing of the airport, allowing monitoring, anomaly detection, and gathering information without physically being present at the site.

10. **Store_results**: task within the workflow orchestration system is responsible for storing and persisting the outputs generated by previous tasks within the workflow.

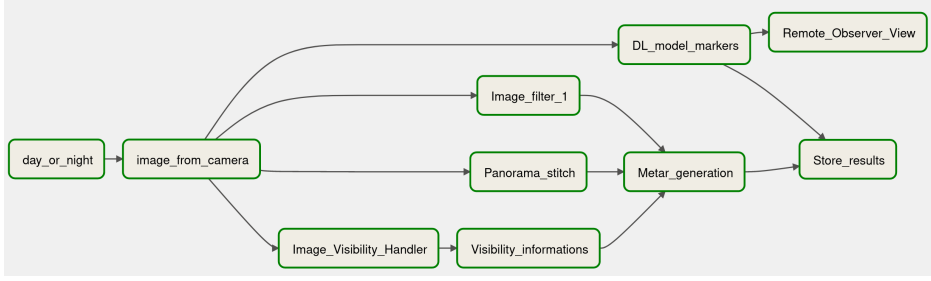


Figure 5. Workflow for the airport meteorology application in AirFlow

One of the key advantages of our Airflow-based workflow orchestration system is the flexibility between data-driven and computation-driven serverless functions. This allows the workflow to efficiently scale the functions based on their utilization, which is a vital aspect of such applications. By dynamically allocating resources to functions as needed, Airflow ensures optimal performance and resource utilization within the workflow.

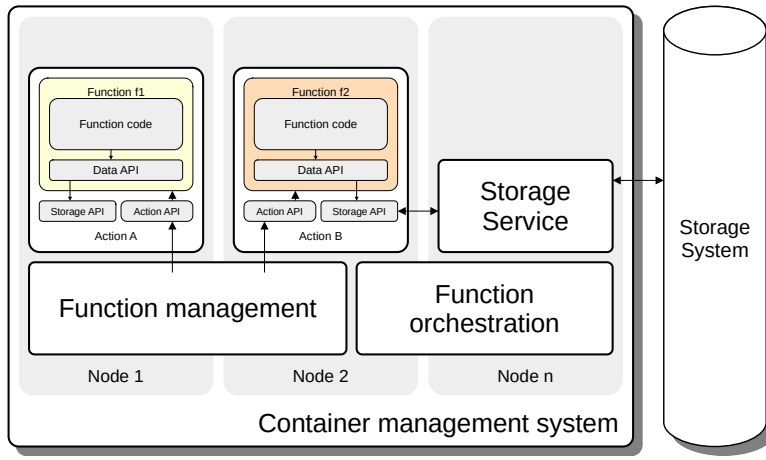


Figure 6. Generic architecture of a Function-as-a-Service application

6 THE ARCHITECTURE OF A FUNCTION-AS-A-SERVICE APPLICATION

Based on our experiments with adapting the airport visibility application to FaaS concepts, we have acquired enough experience to be able to start formulating a generic architecture of a Function-as-a-Service application – see Figure 6.

The whole application is sitting in a cluster of nodes managed by a container management system. We are using the Kubernetes containerization system, but any other system able to manage a cluster of Docker (or others) containers can be used instead of Kubernetes. Inside this cluster is deployed a function management system, able to create parametrized function instances, execute them, provide them with input parameters and communicate their output back to the function caller. In our case we are using the Apache OpenWhisk as the function manager, but OpenFaaS [20] could be also used, for example. To actually create an application out of a group of functions, they need to be orchestrated. For this the architecture containes a function orchestration component, which is able to create workflows of functions, provide them with inputs and store or display their outputs. In our case, we are using the Apache Airflow. Additionally, the cluster contains a deployment of a storage management system. In our case we are using the NextCloud system and its WebDAV server, but this can be replaced by any other cloud storage management, providing other data access protocols – FTP, NFS, SMB and others. The choice of a concrete cloud storage management is dependent on the requirements of the application itself.

Above this software infrastructure are the functions themselves. In the terminology of Apache OpenWhisk, which we have adopted in our research, a prepared

function is called an action. An action can be activated, thus executing the function which it contains on concrete input data. The whole function is executed inside a runtime environment – in the case of Apache OpenWhisk, and any other FaaS framework using containerization, a runtime environment is a pre-prepared Docker container, containing all the necessary software to execute the function code. So the concrete runtime environment is selected based on the characteristics of the function. A generic Apache OpenWhisk installation distinguishes functions only by the programming language in which they are written, and provides runtime environments – Docker images – for functions written in Java, JavaScript, PHP, GO, Python, .NET, Swift and Ruby. It also provides a "black-box" runtime environment for functions not written in a managed programming language. However, this Docker image is just a template, and has to be modified for every such a function.

Each runtime environment provides certain APIs and interfaces which allow the function to interact with the user, the orchestration system, and the storage management system. Most important of these is a REST API receiving the function code and input data, compiling, executing the function code, and returning its result. In the case of Apache OpenWhisk this REST API is provided by a pre-prepared server, which is part of each runtime environment.

Another important part of the function infrastructure is the access to storage management. In our architecture, we have divided the storage management infrastructure available to functions into two layers. The lower layer, named Storage API in Figure 6, is pre-configured with the actual storage end-point information (the server data, so to say). The higher layer, through which the function code accesses data, called Data API in Figure 6, is server-agnostic, or *serverless*. Its structure and method signatures depend on the application's domain, not on the actual underlying storage technology. For example in the case of our application, dealing mainly with large collections of time-coded images, the data are identified by date, time, image origin and image resolution – all metadata. The resulting data structure created in the Storage System and its translation from/to the metadata coordinate system is the responsibility of the Data API, and the function itself is unaware of it.

The runtime environment may contain other APIs or interfaces, depending on the requirements of the application. For example an application which uses a camera system may require that the runtime environment provides an abstract API for accessing these cameras. An application using distributed networked sensors may require an IoT API, and an application using artificial intelligence and machine learning methods will require APIs which will give it access to specialized hardware. All these APIs should follow the pattern used by the storage API – a lower level API configured for the specific external environment (camera network, IoT network, GPU cluster), and a higher-level API used by the function itself, agnostic of the hardware specifics and using only metadata to express task parameters. This will allow the function developer to actually work in a serverless environment.

7 SUMMARY AND FUTURE WORK

In this paper we have described the process of transforming a legacy application into the Function-as-a-Service paradigm. The application is used for measuring visibility at airports. We have described how it has been divided into modules, and how these modules are being transformed into serverless cloud functions. We have also shown the use of OpenWhisk to manage and run those functions (actions in OpenWhisk terminology), and how to manage the actions using Apache AirFlow.

Most importantly, we have been able to derive from this work a partial architecture of such a serverless cloud application. The architecture includes data access, which is not being handled by OpenWhisk or any of the other FaaS tools we use. The architecture of the data access subsystem adheres to the serverless principle – the action code accesses data by their metadata, not by their location.

In our future work, we will continue to transfer more of the application's blocks into the serverless cloud as OpenWhisk functions. We will also extend the generic architecture, to encompass more of the application's components, and if possible include other methods of data storage, like NoSQL [21]. We will try to extract a generalized methodology for the transformation of legacy applications into the serverless cloud paradigm, and use it to transfer another application to which we have access, for example [22].

Another goal of our future work is to try and apply a cloud risk assessment method to the application [23].

Acknowledgements

This publication is the result of the project implementation: Research on the application of artificial intelligence tools in the analysis and classification of hyperspectral sensing data (ITMS: NFP313011BWC9) supported by the Operational Programme Integrated Infrastructure (OPII) funded by the ERDF. It is also supported by APVV grant No. APVV-20-0571 and VEGA grant No. 2/0131/23.

REFERENCES

[1] Chowhan, K.: Hands-on Serverless Computing: Build, Run and Orchestrate Serverless Applications Using AWS Lambda, Microsoft Azure Functions, and Google Cloud Functions. Packt Publishing Ltd., 2018.

[2] Baldini, I.—Castro, P.—Chang, K.—Cheng, P.—Fink, S.—Ishakian, V.—Mitchell, N.—Muthusamy, V.—Rabbah, R.—Slominski, A.—Suter, P.: Serverless Computing: Current Trends and Open Problems. In: Chaudhary, S., Somani, G., Buyya, R. (Eds.): Research Advances in Cloud Computing. Springer, Singapore, 2017, pp. 1–20, doi: 10.1007/978-981-10-5026-8_1.

[3] Marek, K.—Śmiałek, M.—Rybiński, K.—Roszczyk, R.—Wdowiak, M.: BalticLSC: Low-Code Software Development Platform for Large Scale Compu-

tations. Computing and Informatics, Vol. 40, 2021, No. 4, pp. 734–753, doi: 10.31577/cai_2021_4_734.

[4] Sbarski, P.—Cui, Y.—Nair, A.: Serverless Architectures on AWS, Second Edition. Manning, 2020.

[5] Rose, R.: Hands-on Serverless Computing with Google Cloud: Build, Deploy, and Containerize Apps Using Cloud Functions, Cloud Run, and Cloud-Native Technologies. Packt Publishing Ltd., 2020.

[6] Kumar, V.—Agnihotri, K.: Serverless Computing Using Azure Functions: Build, Deploy, Automate, and Secure Serverless Application Development with Azure Functions. BPB Publications, 2021.

[7] Abdel-Aty, M.—Ekram, A. A.—Huang, H.—Choi, K.: A Study on Crashes Related to Visibility Obstruction Due to Fog and Smoke. Accident Analysis and Prevention, Vol. 43, 2011, No. 5, pp. 1730–1737, doi: 10.1016/j.aap.2011.04.003.

[8] Özdemir, E. T.—Deniz, A.—Sezen, İ.—Menteş, Ş. S.—Yavuz, V.: Fog Analysis at Istanbul Ataturk International Airport. Weather, Vol. 71, 2016, pp. 279–284, doi: 10.1002/wea.2747.

[9] Bueso, J.—Rojas Gregorio, J.—Lozano, M.—Pino Gonzalez, D.—Prats, X.—Miglietta, M.: Influence of Meteorological Phenomena on Worldwide Aircraft Accidents in the Period 1967–2010. Meteorological Applications, Vol. 25, 2018, pp. 236–245, doi: 10.1002/met.1686.

[10] Burnham, D. C.—Spitzer, E. A.—Carty, T. C.—Lucas, D. B.: United States Experience Using Forward Scattermeters for Runway Visual Range. U.S. Department of Transportation, Federal Aviation Administration, 1997.

[11] Bartok, J.—Ivica, L.—Gaál, L.—Bartoková, I.—Kelemen, M.: A Novel Camera-Based Approach to Increase the Quality, Objectivity and Efficiency of Aeronautical Meteorological Observations. Applied Sciences, Vol. 12, 2022, No. 6, Art. No. 2925, doi: 10.3390/app12062925.

[12] Bartok, J.—Šišan, P.—Ivica, L.—Bartoková, I.—Malkin Ondík, I.—Gaál, L.: Machine Learning-Based Fog Nowcasting for Aviation with the Aid of Camera Observations. Atmosphere, Vol. 13, 2022, No. 10, Art. No. 1684, doi: 10.3390/atmos13101684.

[13] MicroStep-MIS: IMS4 Remote Observer. https://www.microstep-mis.com/drupal/web/sites/default/files/datasheets/IMS4%20Remote%20Observer_product%20sheet.pdf.

[14] Sciabarra, M.: Learning Apache OpenWhisk. O'Reilly Media, Inc., 2019.

[15] Foundation, T. A. S.: OpenWhisk Incubation Status - Apache Incubator. https://incubator.apache.org/projects/openwhisk.html.

[16] Thomas, J.: OpenWhisk Docker Actions. 2017, https://jamesthom.as/2017/01/openwhisk-docker-actions/.

[17] Thomas, J.: Large (Java) Applications on Apache OpenWhisk. 2019, https://jamesthom.as/2019/02/large-java-applications-on-apache-openwhisk/.

[18] Haines, S.: Workflow Orchestration with Apache Airflow. Modern Data Engineering with Apache Spark: A Hands-on Guide for Building Mission-Critical Streaming

Applications, Apress, Berkeley, CA, 2022, pp. 255–295, doi: 10.1007/978-1-4842-7452-1_8.

[19] Harenslak, B. P.—de Ruiter, J. R.: Data Pipelines with Apache Airflow. Manning, 2021.

[20] Bobák, M.—Hluchý, L.—Tran, V.: Tailored Platforms as Cloud Service. 2015 IEEE 13th International Symposium on Intelligent Systems and Informatics (SISY), 2015, pp. 43–48, doi: 10.1109/SISY.2015.7325408.

[21] Faridoon, A.—Imran, M.: Big Data Storage Tools Using NoSQL Databases and Their Applications in Various Domains: A Systematic Review. Computing and Informatics, Vol. 40, 2021, No. 3, pp. 489–521, doi: 10.31577/cai_2021_3_489.

[22] Krammer, P.—Kvassay, M.—Forgáč, R.—Očkay, M.—Skovajsová, L.—Hluchý, L.—Skurčák, L.—Pavlov, L.: Regression Analysis and Modeling of Local Environmental Pollution Levels for the Electric Power Industry Needs. Computing and Informatics, Vol. 41, 2022, No. 3, pp. 861–884, doi: 10.31577/cai_2022_3_861.

[23] Zbořil, M.: Risk Assessment Method of Cloud Environment. Computing and Informatics, Vol. 41, 2022, No. 5, pp. 1186–1206, doi: 10.31577/cai_2022_5_1186.

Ondrej Habala is a researcher at the Institute of Informatics of the Slovak Academy of Sciences. He works mainly with distributed computing systems and cloud systems, applying them towards solving domain-specific problems mainly in meteorology and hydrology. He has participated in more than 10 national and international research projects, including EU FP5, FP6, FP7, H2020 and HE projects. He is the author of more than 80 publications in his research field.

Martin Bobák works as a researcher at the Institute of Informatics of the Slovak Academy of Sciences. His focus is mainly in cloud computing and cloud architectures. He has participated in several research projects, including FP7, H2020 and HE European research programs. He is the autor of more than 25 scientific publications in the areas of cloud computing, artificial intelligence and data science.

Martin Šeleng is a researcher at the Institute of Informatics of the Slovak Academy of Sciences. He specializes in research infrastructures, cloud computing, and machine learning. He has participated in several research projects, including in the FP5-FP7, H2020 and HE European research programs. He is the author of over 50 scientific publications.

Viet Tran is a senior researcher at the Institute of Informatics of the Slovak Academy of Sciences. His main work is in cloud computing and research infrastructures. He is a senior team leader at II SAS, and has led II SAS researchers in several H2020 and HE European research projects. He is the author of approximately 100 scientific publications.

Ladislav Hluchý is a senior researcher and the head of the Department of Parallel and Distributed Information Processing at the Institute of Informatics of the Slovak Academy of Sciences. He has been active in European research programs since FP4, and has led II SAS team in dozens of research projects in FP4, FP5, FP6, FP7, H2020 and HE. Over his research career he has been the author of over 150 scientific publications. His specialization is distributed information processing, cloud computing and data science.

Computing and Informatics, Vol. 42, 2023, 896–922, doi: 10.31577/cai_2023_4_896

MOOA-CSF: A MULTI-OBJECTIVE OPTIMIZATION APPROACH FOR CLOUD SERVICES FINDING

Youcef Bezza

ICOSI Laboratory, Abbes Laghrour University
Khenchela, Algeria
e-mail: bezza.youcef@univ-khenchela.dz

Ouassila Hioual

Abbes Laghrour University, Khenchela, LIRE Laboratory of Constantine 2
Algeria
e-mail: ouassila.hioual@gmail.com

Ouided Hioual

Abbes Laghrour University, Khenchela, Algeria
e-mail: ouided.hioual@gmail.com

Derya Yiltas-Kaplan, Zeynep Gürkaş-Aydin

Department of Computer Engineering, Istanbul University-Cerrahpaşa
Avcilar, Istanbul, Türkiye
e-mail: {dyiltas, zeynepg}@iuc.edu.tr

Abstract. Cloud computing performance optimization is the process of increasing the performance of cloud services at minimum cost, based on various features. In this paper, we present a new approach called MOOA-CSF (Multi-Objective Optimization Approach for Cloud Services Finding), which uses supervised learning and multi-criteria decision techniques to optimize price and performance in cloud

computing. Our system uses an artificial neural network (ANN) to classify a set of cloud services. The inputs of the ANN are service features, and the classification results are three classes of cloud services: one that is favorable to the client, one that is favorable to the system, and one that is common between the client and system classes. The ELECTRE (ÉLimination Et Choix Traduisant la REalité) method is used to order the services of the three classes. We modified the genetic algorithm (GA) to make it adaptive to our system. Thus, the result of the GA is a hybrid cloud service that theoretically exists, but practically does not. To this end, we use similarity tests to calculate the level of similarity between the hybrid service and the other benefits in both classes. MOOA-CSF performance is evaluated using different scenarios. Simulation results prove the efficiency of our approach.

Keywords: MCDM, cloud computing, optimization, artificial neural networks, genetic algorithm, similarity measures, supervised learning

1 INTRODUCTION

In recent years, users have become increasingly accustomed to using the internet to obtain software resources. This is done in the form of web services, provided by information technology organizations, and can be accessed by end users over the internet [1]. Cloud computing is a service delivery paradigm that provides access to services and resources. It is defined by the National Institute of Standards and Technology as a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications, and services) that can be quickly provisioned and released with minimal management effort or service provider interaction [2]. Cloud computing has three service models: Software as a Service, Platform as a Service, and Infrastructure as a Service. Furthermore, it has four deployment models: private, public, hybrid, and community [3]. The cloud service provider (CSP) supplies services to users as a rental. Due to the huge number of available virtual cloud resources, the CSP role is very complex. As such, researchers have given more attention to cloud service performance [3].

With the development of cloud computing technology, a single web service can no longer meet users' needs, since these are often complex. On the other hand, since different service providers offer web services with the same functionalities, but different in terms of their criteria, selecting the best web service that satisfies user needs is a difficult problem. This can decrease the performance of cloud services, which has a direct impact on the client's business. So, the more cloud performance is optimized, the more client confidence is increased [4].

Performance optimization of cloud computing is about making the components in the cloud meet component-level requirements and client expectations. We aim to increase the performance of cloud services with a minimum cost, depending on various constraints. Performance optimization allows us to improve the performance

of various factors [4]. To meet the needs of different clients, it is important to optimize the performance of cloud computing. In the literature, some solutions have been proposed. Among these solutions, the approaches can be cited based on: hybrid optimization [5, 6], genetic algorithm [7, 8, 9], and multi-objective optimization [3, 10, 11].

They are classified into two categories:

1. Approaches that optimize cloud performance on the client's side, in which response time and cost factors are always taken into account as users always need the best services at the lowest cost and response time; however, the cost factor is equally crucial. Since the service cost is a major factor to provide QoS, especially for commercial customers, therefore, compromising the response time for long deadline requests is desirable compared to the compromising in the cost factor [12].
2. Approaches that optimize cloud performance on the system side apply various techniques and strategies, such as caching, compression, load balancing, autoscaling, and serverless computing. This allows reduced latency, increased throughput, enhanced availability, and resources savings. To our knowledge, very few works that have taken into account user needs and system performance.

To address these challenges, this paper proposes a novel approach to optimize cloud computing performance. Compared to the above approaches, our proposition takes into consideration both the optimization of user and system preferences. Therefore, our contribution allows to solve the problem of the cloud service finding while satisfying user and system constraints. Our approach is based on neural network classification, multi-criteria decision-making systems (MCDM), optimization algorithms (genetic algorithm), and similarity measurements. First, we use an artificial neural network (ANN) to classify cloud services. The ANN inputs are service criteria. The classification results are three classes of cloud services. The first class is composed of services that are favorable for the client side (i.e. that satisfy client needs). The second class is composed of services that are favorable for the cloud provider side. In addition, the third one contains services that are common between the client and the provider. Concerning the MCDM, we have chosen to use the ELECTRE method. It is applied to sort services in each class. The purpose of this sorting is to eliminate weak services from the three classes. In a second step, we have modified the genetic algorithm (GA) to make it adaptive to our system. The genetic algorithm result is a hybrid cloud service that theoretically exists but practically does not. For this, we use similarity tests to calculate the level of similarity between the hybrid service and the other services to obtain the best service that meets the client's needs at a low price.

The rest of this paper is organized as follows: background and preliminaries are presented in Section 2, related work in Section 3, proposed model in Section 4, case study in Section 5, experimental results and evaluation in Section 6, and conclusion in Section 7.

2 BACKGROUND AND PRELIMINARIES

In this section, we will first give a brief overview of ANN, followed by an introduction to the ELECTRE II method that we consider in our proposal.

2.1 Artificial Neural Network

ANNs are an information-processing paradigm that simulates the behavior of the human brain for a specific task or function [13]. This type of network is composed of several sets of calculations called neurons, which are combined in layers and operate in parallel. The information is propagated from the input layer to the output layer.

ANNs can store empirical knowledge and make it available to users. The knowledge of the network is stored in synaptic weights, obtained through the process of adaptation or learning [14]. Activation values are transmitted from neuron to neuron based on the weights and activation functions. Each neuron adds up the activation values it receives and then changes the value according to its activation function. The activation procedure follows a look-ahead process and the difference between the predicted value and the actual value (error) is propagated backward by distributing it among the weights of each neuron according to the amount of error for which its neuron is responsible [14].

2.2 ELECTRE II

ELECTRE II [15] is a multi-criteria analysis method that solves decision problems with greater accuracy. This method was the first of the ELECTRE methods specifically designed to deal with ranking problems. The evaluation matrix is the starting point of the ELECTRE II method, in which alternatives are evaluated on different criteria. It aims to rank actions from best to worst. Based on a total pre-ordering principle, ELECTRE II assumes that all actions are comparable; incomparability is excluded, i.e., the decision-maker can always choose between action A and action B.

ELECTRE consists of two main steps. The first step is the preparation of the decision matrix (see Table 1), where g_{ij} denotes the value of variant i with respect to criterion j. The second step is the calculation of the concordance and discordance matrices [15].

		Criteria			
		C_1	C_2	...	C_n
Alternatives	a_1	g_{11}	g_{12}	...	g_{1n}
	a_2	g_{21}	g_{22}	...	g_{2n}
	...	...	...	...	...
	a_n	g_{n1}	g_{n2}	...	g_{nn}

Table 1. Decision matrix illustration

2.2.1 Calculation of the Concordance Matrix

The concordance matrix is generated by summing the weights of the elements in the concordance set. The strength of the hypothesis that alternative A_i is at least as good as alternative A_j is evaluated usisng the concordance index between the pair of alternatives A_i and A_j which is calculated using formula (1) [16]:

$$c(a,b) = \frac{\sum k_j}{k}, \quad \text{where } g_{j(a)} \geq g_{j(b)} \forall_j, \tag{1}$$

where $g_{j(a)}$ and $g_{j(b)}$ are the sets of criteria for which a is equal or preferred to b, k_j is the weight of the j^{th} criterion.

2.2.2 Calculation of the Discordance Matrix

The discordance index D (a, b) is calculated by formula (2) or (3):

$$D(a,b) = 0, \quad \text{if } \forall j, g_{j(a)} \geq g_{j(b)} \tag{2}$$

else

$$D(a,b) = \frac{1}{\sigma} \text{MAX} j[g_{j(b)} - g_{j(a)}], \tag{3}$$

$$\sigma = max \left| g_{j(b)} - g_{j(a)} \right|. \tag{4}$$

After calculating the concordance and discordance indices for each pair of alternatives, two types of outranking relationships are constructed by comparing these indices with two pairs of threshold values: $(C+, D+)$ and $(C-, D-)$. The pair $(C+, D+)$ is defined as the concordance and discordance thresholds for the strong outranking relationship, and the pair $(C-, D-)$ is defined as the thresholds for the weak outranking relationship, where $C+ > C-$ and $D+ > D-$. Then, outranking relationships are constructed according to the following two rules [15]:

- If $C(a,b) \geq C+$, $D(a,b) \leq D+$ and $C(a,b) \geq C(b,a)$, then alternative a is considered to strongly outperform alternative b. Likewise,
- If $C(a,b) \geq C-$, $D(a,b) \leq D-$ and $C(a,b) \geq C(b,a)$, then alternative a is considered to weakly outperform alternative b.

The values of $C-$, $C+$, $D-$, $D+$ are given by the decision makers [15].

3 RELATED WORK

The performance optimization of cloud computing, based on different features, has become increasingly important in today's world. For this reason, there are many studies developed in the literature. In this section, we will focus on some of them and suggest a comparative table (see Table 2) to introduce an analysis of these studies based on different parameters (features).

In [17], Guo et al. proposed a queuing model and developed a synthetic optimization method to optimize the performance of services. They analyzed and conducted the equation of each parameter of the services in the data center. Then, by analyzing the queuing system's performance parameters, they proposed the synthetic optimization mode, function, and strategy. Finally, they set up the simulation based on the synthetic optimization mode. By comparing and analyzing the simulation results to classical optimization methods, the authors showed that the proposed model can optimize the average wait time, average queue length, and number of clients.

The authors in [18] proposed a prediction-based dynamic multi-objective evolutionary algorithm, named NN-DNSGA-II. They incorporated an ANN with the NSGA-II. The optimization objectives taken into account included minimizing makespan, cost, energy, and imbalance, while maximizing reliability and utilization. The authors demonstrated that in Dynamic Multi-objective Optimization Problems (DMOPs) with unknown true Pareto-optimal fronts, the NN-DNSGA-II algorithm showed remarkable superiority over other alternatives. It outperformed them in various metrics, such as the number of non-dominated solutions, Schott's spacing, and the Hypervolume indicator in most cases.

The authors of [19] expanded the functionality of an existing parallel software framework called WoBinGO, which was initially designed for GA-based optimization, to be suitable for deployment in a cloud environment. Additionally, the researchers introduced an intelligent decision support engine that utilizes artificial neural networks (ANN) and metaheuristics. This engine enables users to evaluate the framework's performance on the underlying infrastructure concerning optimization duration and resource consumption cost. By conducting this assessment, users can make informed decisions based on their preferences, whether they prioritize faster result delivery or lower infrastructure expenses.

Authors of [20] recognized the role of the innovative Grasshopper Optimization Algorithm (GOA). They have strongly highlighted the significance of such an algorithm for optimizing resource allocation in a cloud computing environment. The proposed algorithm was simulated with MATLAB using eight datasets. Furthermore, the authors conducted a comparative analysis between the Grasshopper Optimization Algorithm (GOA) and the genetic algorithm (GA) and SElf-adaptive Inertia weight and Random Acceleration (SEIRA) algorithms. This comparison aimed to accurately assess the performance of GOA. The findings demonstrated the effectiveness of the proposed GOA in efficiently solving the resource allocation problem in the cloud.

In [21], Salem et al. created a new algorithm (MOABC) derived from a combination of ABC and Multi-Objective Optimization. Hence, the study introduced optimized replica placement strategies to determine the most suitable locations based on minimum distance and cost-effective paths. Additionally, for direct bees, the approach focused on identifying the shortest routes in terms of distance and lower cost. The proposed algorithm gave fast access to data and selected the best replica placement nearest to users. Additionally, the placement optimization pro-

vided more least-cost paths, better response times, and replication costs within the budget.

Authors of [22] used a hybrid metaheuristic algorithm, namely, the Whale Optimization Algorithm (WOA) with Simulated Annealing (SA), to optimize the energy consumption of sensors in IoT-based WSNs. To simulate the IoT network, the authors used the Xively IoT platform. Several performance metrics, such as load, residual energy, number of alive nodes, cost function, and temperature, were used to choose the optimal CHs in the IoT network. The proposed work in [22] was subjected to a comparison with various state-of-the-art approaches, and it demonstrated favorable results.

In [23], the authors introduced a new and innovative approach for performance optimization using a multi-agent system, which is based on both the Internet of Things (IoT) and the deep learning paradigm. They took advantage of the state-of-the-art probabilistic, recurrent neural network, and long short-term memory models to predict, intelligently, the upcoming behavior and optimization needs of the system. They deployed the proposed performance optimization approach and showed significant performance gain in comparison with existing approaches.

In [5], a hybrid optimization model has been developed which allows for an efficient task allocation to the virtual machines (VMs) in cloud computing. The task priorities are managed by using the hierarchy process. The authors have used BAT (Bandwidth-aware divisible task) and BAR models to consider the task properties and VM characteristics for task scheduling. They also used MOML (the minimum overload and minimum lease) preemption policy which was successfully employed to reduce the load on the VMs. The performance of the proposed model was then compared with existing algorithms such as BAT and ACO (ant colony optimization) algorithms. Consequently, the authors have been able to prove that their model is efficient in terms of resource, bandwidth, and memory utilizations.

In [24], the authors proposed a decision support engine that recommends optimal framework parameters to achieve minimal total execution time and total cumulative uptime for a specified optimization problem. The engine solves a bi-criteria optimization problem and uses surrogate models of the IaaS behavior under various large-scale optimization loads as a fitness evaluator in MOGA (multi-objective genetic algorithms). According to the authors, the obtained results were promising, especially in the case of computationally heavy fitness evaluation functions.

In [25], the authors introduced a novel approach named Multifaceted Optimization Scheduling Framework (MFOSF), which integrates scheduling and resource cost chronology models. According to the authors, this framework effectively illustrates the relationship between the user's budget and the producer's cost during the planning process.

In [26], Zhou et al. proposed a cloud service optimization method based on an artificial ant colony algorithm and bee colony algorithm (DAABA). To enhance the applicability of the farming season, the authors incorporated both the dynamic coefficient strategy and the reliability feedback update strategy into the optimization model. These additions were made to strengthen the overall performance and

adaptability of the model. Furthermore, the optimal fusion evaluation strategy was used to save optimization time by reducing useless iterations, while the iterative adjustment threshold strategy was adopted to improve the accuracy of cloud service finding by increasing the size of the bee colony.

Ragmani et al. [27] proposed a hybrid Fuzzy Ant Colony Optimization (FACO) algorithm for VM scheduling to guarantee high efficiency in a cloud environment. The proposed fuzzy module evaluates historical information to calculate the pheromone value and select a suitable server while keeping an optimal computing time [27]. Their study provides one of the first investigations into how to choose the optimal parameters of ant colony optimization algorithms using the Taguchi experimental design.

In [28], the authors introduced a hybrid meta-heuristic algorithm based on Firefly Optimization Algorithm and GA to optimize task scheduling in the cloud computing platform for multiple tasks. The developed system provides a distribution by reallocating the loads to the related VMs, taking into account the objective function of the VM. The system, also, increases resource utilization and communication cost during task scheduling and efficiently decreases the processing time of the process compared to different techniques such as GA, Firefly Algorithm, and Modified Firefly Optimization Algorithm.

In their paper [29], the authors proposed a novel hybrid load balancing model based on optimizing a modified particle swarm algorithm. This model incorporates enhanced metaheuristic firefly algorithms, which significantly improve the overall performance of cloud computing systems. The proposed approach primarily focuses on predictive workload allocation, emphasizing resource scalability and implementing a load balancing model that maximizes the utilization of uniformly load-distributed virtual machines (VMs) [29].

The authors of [7] proposed an approach to improve the capability of data centers. It allocates requests among VMs in an inefficient manner by using their current status in cloud computing with a GA [7]. The authors claim that their algorithm uses a modified GA-based approach, in which the best VM is selected by analyzing candidates which have more fitness compared to others. The proposed approach significantly reduced the response time of servers and provided an effective load balancing among VMs [7].

In [30], an efficient optimization method for task scheduling was presented. It is based on a hybrid Multi-Verse Optimizer with a GA (MVO-GA). MVO-GA was proposed to enhance the performance of tasks transfer via the cloud network, based on cloud resources' workload. The proposed method works on multiple properties of cloud resources, namely: speed, capacity, task size, number of tasks, number of VMs, and throughput. The proposed method successfully optimized the task scheduling of a large number of tasks [30]. Also it optimized the large cloud tasks' transfer time, reflecting its effectiveness.

After analyzing the articles cited in Table 2, we can classify them into two main classes. The first class optimizes cloud performance on the client's side [17, 19, 20, 21, 22, 23, 5, 10, 8]. The second class contains articles that optimize cloud perfor-

Research Paper	Environment	Used Parametres					Client/System Side	
		R	Av	C	Th	Rt	Client	System
[17]	Cloud Computing	−	−	−	−	+	+	−
[18]	Cloud Computing	+	−	−	+	+	−	+
[19]	Cloud Computing	−	−	−	−	−	+	−
[20]	Cloud Computing	−	−	−	+	−	+	−
[21]	Cloud Computing	−	−	+	+	−	+	−
[22]	IoT Network	−	−	−	+	−	+	−
[23]	IoT	−	−	−	−	−	+	−
[5]	Cloud Computing	−	−	−	−	+	+	−
[24]	Cloud Computing	−	−	−	−	+	+	−
[25]	Cloud Computing	−	−	−	+	−	−	+
[26]	Cloud Computing	+	−	−	−	−	+	−
[27]	Cloud Computing	−	−	−	−	−	−	+
[28]	Cloud Computing	−	−	−	−	−	−	+
[29]	Cloud Computing	−	−	−	−	−	−	+
[7]	Cloud Computing	−	−	−	−	+	−	+
[30]	Cloud Computing	−	−	−	−	−	−	+
Our Approach	Cloud Computing	+	+	+	+	+	+	+

Table 2. A comparative summarization of some previous studies on performances optimization

mance on the system side [7, 18, 25, 27, 28, 29, 30]. In our research, we introduce a model that optimizes cloud performance for both clients and systems, positioning our proposal at the intersection of these two classes (as shown in Figure 1).

In the model, R, Av, C, Th, and Rt represent Reliability, Availability, Cost, Throughput, and Response Time, respectively.

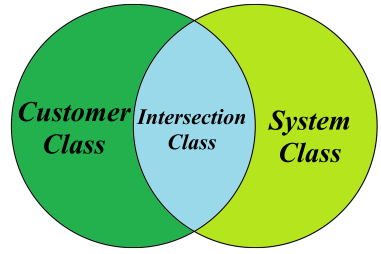


Figure 1. The contextual situation of our research work

4 PROPOSED APPROACH

The successful development of cloud computing has attracted more and more people and companies to use it. On the one hand, the use of cloud computing reduces

costs; on the other hand, it improves efficiency. As users are largely concerned by the quality of services, optimizing the performance of Cloud computing has become essential for its successful application [17]. Furthermore, the number of cloud providers is increasing rapidly. Therefore, the challenge of choosing the cloud provider that best meets a client's needs and optimizes both cost and performance has become a big challenge. In this context, we propose an approach that helps clients to choose the best provider that meets their needs and optimizes both cost and performance.

In this study, we consider a service with five criteria. Two criteria are specifically related to the client side, two criteria are focused on the cloud provider side, and there is one criterion that is common to both the client and the provider. The common criterion in our study is subject to opposing objectives: the client seeks to minimize it, while the provider aims to maximize it. This is precisely why we opted for multi-criteria decision-making systems. In many decision-making problems, diverse perspectives are encountered, often leading to contradictions and differing viewpoints.

4.1 Architecture and Functioning of Our Approach

Figure 2 depicts the overall architecture of our system, which comprises five interconnected components.

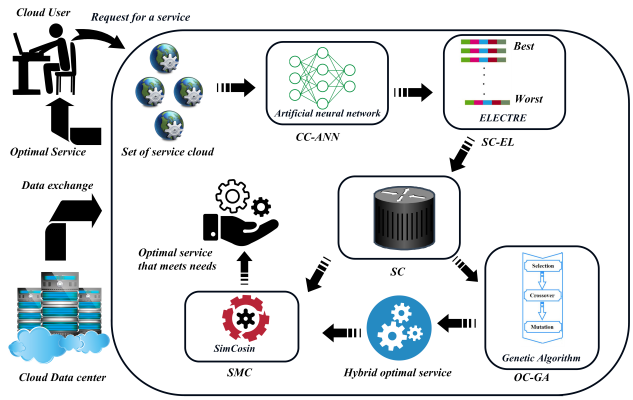


Figure 2. An overview of the proposed general architecture

The first component in our system is the Classification Component (CC-ANN), which utilizes a multi-layer neural network. This component comprises three layers. The first layer is the input layer, where the inputs represent the service criteria. Consequently, we extract the parameters of each cloud service, considering the client's requirements, and represent each service with a vector. The second layer is the hidden layer, which contains the function responsible for making the classification.

Lastly, the third layer is the output layer, generating outputs into three classes: the Client Preferences Class (CPC) containing services preferred by the client, the System Preferences Class (SPC) comprising services preferred for the system, and the Common Services Class (CSC).

The second component in our architecture is referred to as the Sorting Component (SC-El). It relies on the ELECTRE method to arrange the services within each class, ranking them from the best to the worst. This sorting process involves assigning weights to each criterion based on its significance. As for the Storage Component (SC), it serves as a centralized repository within the system, responsible for storing all relevant data.

The fourth component is the Optimization Component (OC-GA), which operates based on the principles of genetic algorithm (GA). Its primary function is to generate a new generation of cloud services derived from the initial three classes mentioned earlier.

The services produced by GA are considered hybrid services, which theoretically exist but are not practically realized. To address this, we need to assess the similarity between these hybrid services and the existing ones. To achieve this, we have introduced the fifth component, the Similarity Component (SMC). The SMC assists in identifying the best service (the closest match) among the services obtained in the preceding steps. The entire process is illustrated in the sequence diagram shown in Figure 3.

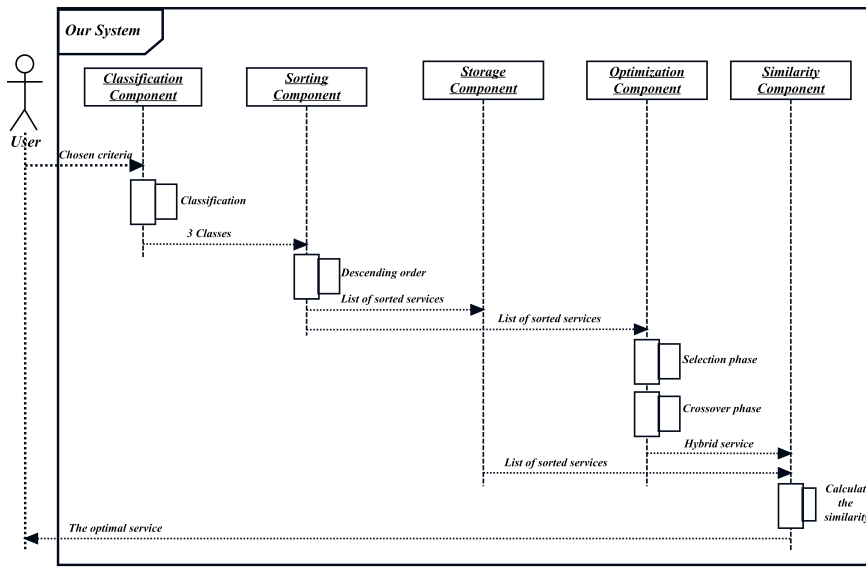


Figure 3. Sequence diagram of the MOOA-CSF functionality

4.2 Classification Step

This step is performed by the CC-ANN component, as shown in Figure 4. The ANN is composed of three layers: input, hidden and output layers. The input layer is comprised of five nodes, denoting the criteria of a service, namely Reliability, Throughput, Availability, Cost, and Response Time. These criteria values are then propagated to the hidden layer. Within the hidden layer, the activation values are passed from neuron to neuron, where each neuron aggregates the received activations and updates its value using a transfer function. This process involves an anticipation mechanism. Subsequently, the difference between the predicted value and the actual value (error) is propagated backward through the network.

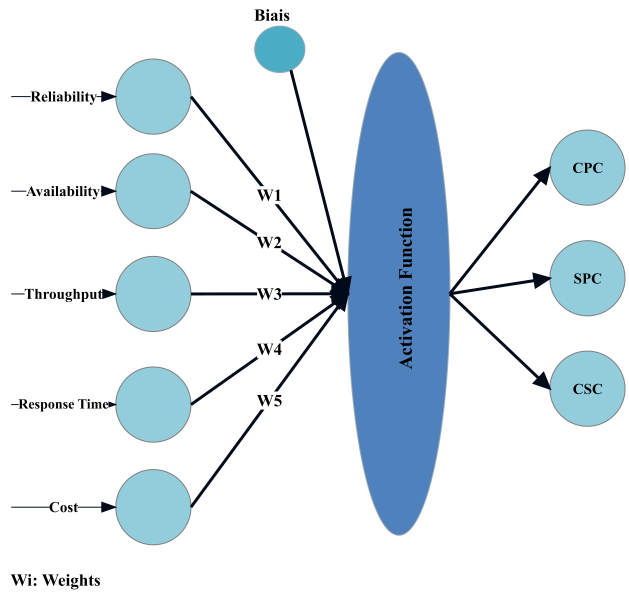


Figure 4. The layers of the CC-ANN component

4.3 Activation Function

The activation function transfers the input values to an output signal. In this paper, we have chosen the hyperbolic tangent function (Tanh). The Tanh function is similar to the sigmoid function, but it is symmetrical around the origin. This results in different signs of outputs from the previous layers being fed into the input of the next layer, as defined by formula (5).

The Tanh function is continuous and differentiable, with values ranging between -1 and 1. Compared to the sigmoid function, the gradient of the Tanh function

is steeper. Tanh is more commonly used than the sigmoid function because it has gradients that are not bounded to vary in a certain direction, and it is also centered on zero [31].

$$Tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}}, \tag{5}$$

where a is the value of the neuron. Figure 5 shows the process of the classification component, where d denotes the desired goal and E is the error margin.

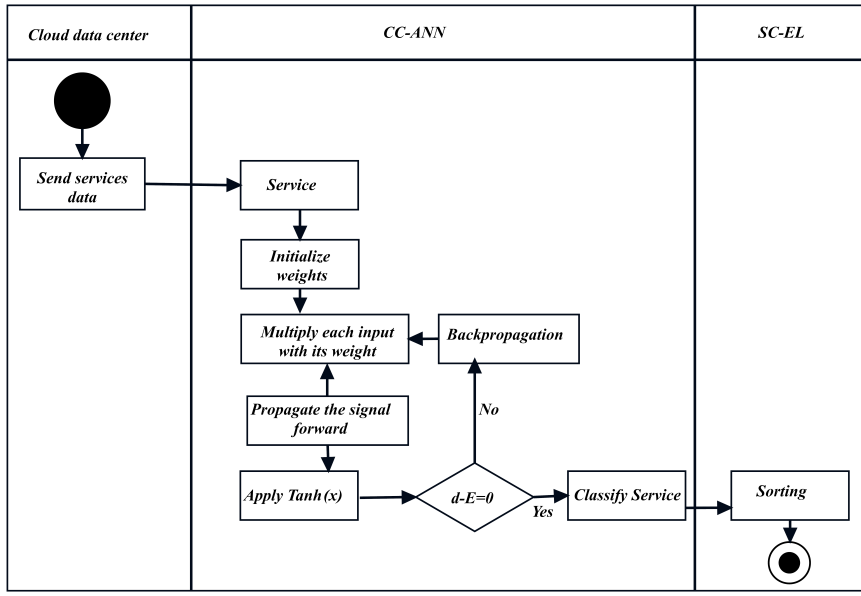


Figure 5. Activity diagram of the CC-ANN component

4.4 Sorting and Elimination Step

This component performs according to the ELECTRE II principle, and is composed of two steps; sorting and elimination. We have to prepare the decision matrix; the alternatives in our case illustrate the services. Each service has five criteria, and each class has a decision matrix that is different from the others.

In the first step, we sort the services of each class from the best to the worst by calculating the concordance, discordance, and dominant matrices. Service a is better than service b if the following strong outranking relation holds:

- If $C(a,b) \geq C+$, $D(a,b) \leq D+$ and $C(a,b) \geq C(b,a)$, then service a is considered to strongly outperform service b.

In the second step, we eliminate the services that are weakly outranking. We consider that a service a is weakly outranking with b when the following condition is true:

- If $C(a,b) \geq C-$, $D(a,b) \leq D-$ and $C(a,b) \geq C(b,a)$, then service a is considered to strongly outperform service b.

4.5 Optimization Step

This component is based on the principle of GA, a metaheuristic approach to solve multi-objective optimization problems. GA is inspired by the principles of Darwin's theory of evolution and is often used as an evolutionary computational model in various fields of study [32]. Currently, GAs are recognized as a very powerful tool in optimization, having been applied in computer science, engineering, education, and stock market data mining optimization [32].

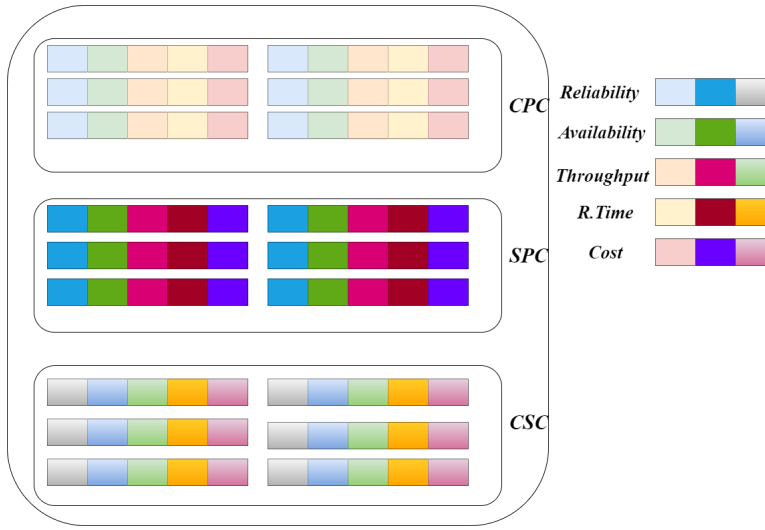


Figure 6. Initial populations of CP, SP and CS classes

We have applied modifications to the GA to make it adaptive to our context. Its operation after modification is as follows:

1. The initial population in our approach is not generated randomly. Instead, we utilize the three classes obtained during the classification phase as our initial populations. Consequently, there are three separate initial populations, not just one (as shown in Figure 6).
2. The population is evaluated by assigning a fitness value to each service, so we can generate a new population.

3. The algorithm determines the termination of the search process based on specific predefined conditions. Typically, these conditions are met when the algorithm reaches a fixed number of generations or when it discovers a satisfactory solution.
4. In case the termination condition is not satisfied, the population proceeds with the selection step. During this step, one service is chosen from each class based on its fitness score, with higher fitness scores leading to a higher likelihood of selection.
5. Following the selection step, the algorithm proceeds to implement crossover on the chosen services, as illustrated in Figure 7. This stage involves creating new services for the subsequent generation through the process of crossing or recombination.

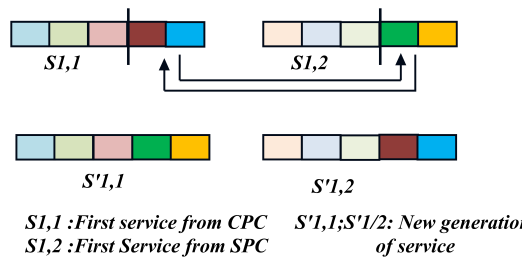


Figure 7. Illustration of the crossover operation results

6. At this stage, the new population returns to the assessment step and the process begins again. We call each cycle of this loop a *generation*.
7. When the termination condition is met, the algorithm breaks out of the loop and usually returns its final search results to the client/provider.

4.6 Similarity Step

The calculation of the degree of similarity between two services is ensured by the SMC, which is based on the similarity SimCosin(X, Y). We suppose that we have two services X and Y represented by two vectors, each vector containing five criteria [33]:

$$X = [R_x, Av_x, C_x, Th_x, Rt_x];$$

$$Y = [R_y, Av_y, C_y, Th_y, Rt_y]$$

Additionally, the function $SimCosin(X, Y)$ must satisfy the following properties [33]:

Property 1: $0 \leq \cos(X, Y) \leq 1$.

Property 2: $\cos(X, Y) = \cos(Y, X)$.

Property 3: $\cos(X, Y) = 1$; if $X = Y$.

5 CASE STUDY

To validate our system, a simulation environment is established. The simulation context proceeds as follows: we suppose that we have several cloud services providers and each one provides a service. A client searches for optimal services that meet their needs among these services, and each service has (m) criteria.

The simulation environment is a PC with the following configuration: Nvidia GeForce GTX 1060 GDDR5, Intel Core i7-7700HQ CPU 2.80 GHz and RAM 16 GB. The programming environment is Eclipse IDE 2020-09. The test data is based on the QWS2 dataset [34] where the number of services is 4000. The neural network was trained using 2600 services. Each service is composed of five criteria.

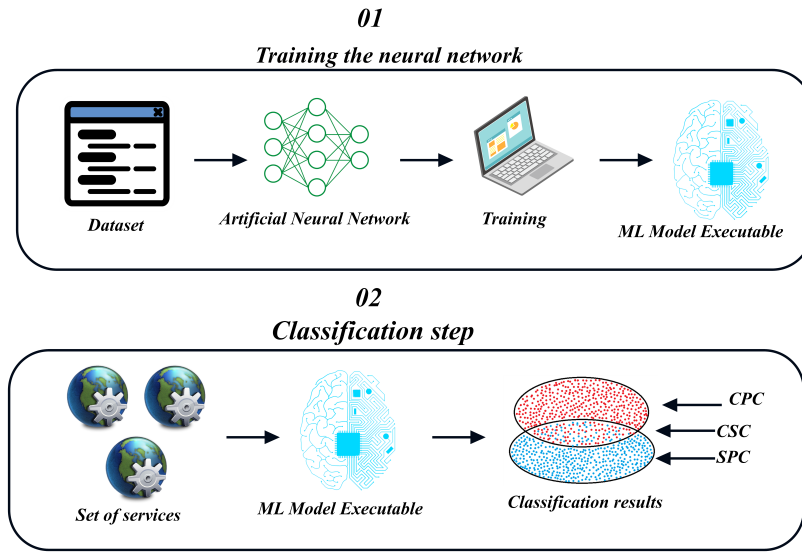


Figure 8. Classification process of the CC-ANN component

As a first phase, we develop a neural network to classify these services. Thus, we assign each service to the appropriate class.

As shown in Figure 8, in the first step we train the neural network using a dataset. We initialize our weights randomly then feed-forward the values from one layer to the next. If the output is not equal to the desired value, we back-propagate from the output neuron to the input neuron. We update the weights and feed-forward the values. We repeat this process until we find the desired values and obtain a model of a neural network.

In the Classification step (Figure 8), we use the model of the neural network to classify services into three categories. Each class contains a set of services that have the same range of values. The first class (CPC) contains 916 services, the second (SPC) contains 640 services, and the last one (CSC) contains 444. After that, we

make a copy list of these services and send it to the storage component. Then, we transfer the result to the sorting component.

As a second phase, the sorting component uses the ELECTRE method to sort and then eliminate services to find the best one in each class. The decision problem is characterized by five parameters or criteria. All criteria are advantage criteria, i.e., performance is better when the score is high.

The weights of the criteria are presented in Table 3.

Criterion	Availability	Cost	Response Time	Throughput	Reliability
Weight	2	3	3	2	1

Table 3. Initialisation of the five criteria weights

Due to the large number of services in the dataset, we will take as a sample the services S6, S8, S9, S907, S908 and S910. Therefore, the service performance matrix of the first class, as illustrated in Table 4, is used to calculate the concordance matrix $C(a, b)$, which is illustrated in Table 5. This is calculated using the formula (1) that was quoted in the previous section.

Service ID	Reliability	Cost	Response Time	Throughput	Availability
Service 6	61	68	1 046	2 178	68
Service 8	73	63	1 250	2 144	75
Service 9	81	124	1 244	2 060	85
Service 907	64	62	1 175	1 655	43
Service 908	73	63	1 188	1 963	50
Service 910	52	110	1 107	1 999	60

Table 4. Performance matrix of the illustrative example

Service ID	Service 6	Service 8	Service 9	Service 907	Service 908	Service 910
Service 6	**1**	0.454	0.181	0.545	0.545	0.454
Service 8	0.545	**1**	0.454	**1**	**1**	0.727
Service 9	0.818	0.545	**1**	**1**	**1**	**1**
Service 907	0.454	**0**	**0**	**1**	**0**	0.454
Service 908	0.454	0.454	**0**	**1**	**1**	0.454
Service 910	0.545	0.272	**0**	0.545	0.545	**1**

Table 5. The concordance matrix relating to the illustrative example

We calculate the discordance matrix $D(a, b)$ of our illustrative example using formula (3), results are shown in Table 6. To obtain σ, we calculate, for each attribute, the differences between all its values in the dataset. Moreover, the attribute corresponding to the maximum value of these deviations is maintained. We then calculate σ, which is equal to the maximum value of the maintained attribute minus its

minimum value (see formula (4)). According to the QWS2 dataset, the maintained attribute is the Cost attribute consequently $\sigma = 140$.

Service ID	Service 6	Service 8	Service 9	Service 907	Service 908	Service 910
Service 6	**0**	0.085	0.4	0.021	0.081	0.3
Service 8	0.035	**0**	0.435	**0**	**0**	0.335
Service 9	0.0008	0.004	**0**	**0**	**0**	**0**
Service 907	0.178	0.228	0.442	**0**	0.064	0.342
Service 908	0.128	0.178	0.435	**0**	**0**	0.335
Service 910	0.064	0.15	0.207	0.085	0.15	**0**

Table 6. The discordance matrix relating to the illustrative example

After calculating the concordance and discordance matrices, the dominant matrix is constructed from the two concordance and discordance indices. Thus, the following strong and weak outranking indices are obtained: $C+ = 0.850$, $C- = 0.750$, $D+ = 0.200$, and $D- = 0.300$. We obtain the dominant matrix as illustrated in Table 7.

Service ID	Service 6	Service 8	Service 9	Service 907	Service 908	Service 910
Service 6	Strong	–	–	–	–	–
Service 8	–	Strong	–	Strong	Strong	–
Service 9	Weak	–	Strong	Strong	Strong	Strong
Service 907	–	–	–	Strong	–	–
Service 908	–	–	–	Strong	Strong	–
Service 910	–	–	–	–	–	Strong

Table 7. The dominant matrix relating to the illustrative example

Figure 9 shows the final rank between services, to get this graph we follow this two properties:

- If service a outranks service b, an arrow starting at vertex a and ending at vertex b.
- If no outranking relation exists between the two services a and b, then no arrow can be drawn between the two vertices.

After that, we sort services to eliminate those with weak outclass relations. A service with a weak outclass relation to another service means that the former is included in the latter. The remaining services, which have not been eliminated, are the best in each class. We use these services as an initial population. This latter is used to generate new populations based on the good services. We repeat the same process for the second and third classes, then transferring the resulting services to the optimization component. The OC will consider each class as an initial population. At first, the OC crosses a service from the first class with the services from the second class. Then, we evaluate the new service. If the value is less than

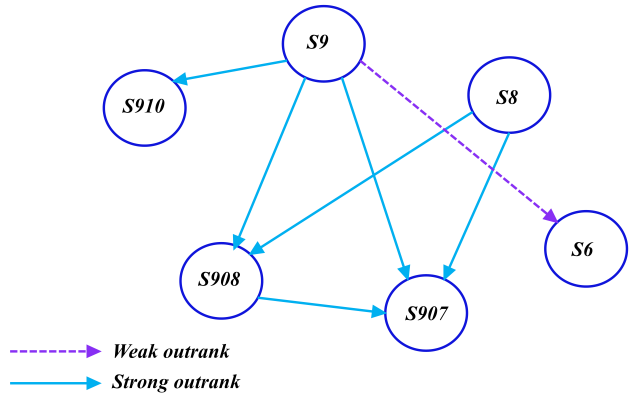


Figure 9. The outrank relation between services relating to the illustrative example

the fitness function, we eliminate the service. After crossing all services of the first class with all services of the second class, we will obtain a new population. Then we cross a service from the second class with the services of the third class, obtaining a second population. After that, we cross the services of the new populations to get the final one.

The final population contains hybrid services; the latter are abstract. To design the optimal service, we must calculate the similarity between services already stored in our storage component and those of the final population.

The similarity component (Figure 10) calculates the similarity index between the best services of each class and services of the final population. In our case, we use the Sim-Cos function. If the similarity index is in the interval [0.8:1], we keep the service. Furthermore, we keep all services that have a high index. Then, we send the list of these services to the client.

6 EXPERIMENTAL RESULTS AND EVALUATION

In this section, we evaluate our system based on the number of services in each class. We assume that there are more than 100 services in each class. In the experiments (1, 2, and 3), the values of response time and throughput do not change because they include technical aspects (servers, computer network, etc.) as well as those related to the interface ergonomics between the user and the system.

6.1 Experiment 1

The goal of this experimentation is to show the average of services in the client-preferred-class (CPC) before and after optimization. As shown in Figure 11, the values of response time, throughput, availability, reliability, and cost before optimization are, respectively, 1.12, 1.6, 83 %, 89 % and $ 110. After optimization,

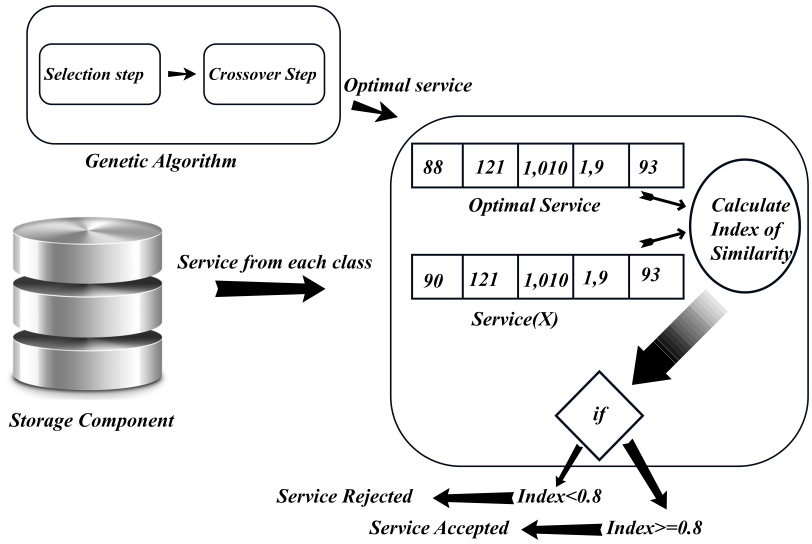


Figure 10. The similarity component functioning

availability increases from 83 % to 92 %, reliability from 89 % to 96 % and cost decreases from $ 110 to $ 93. Through this experimentation, we note that the values of three criteria have been optimized, due to the number of crossed services.

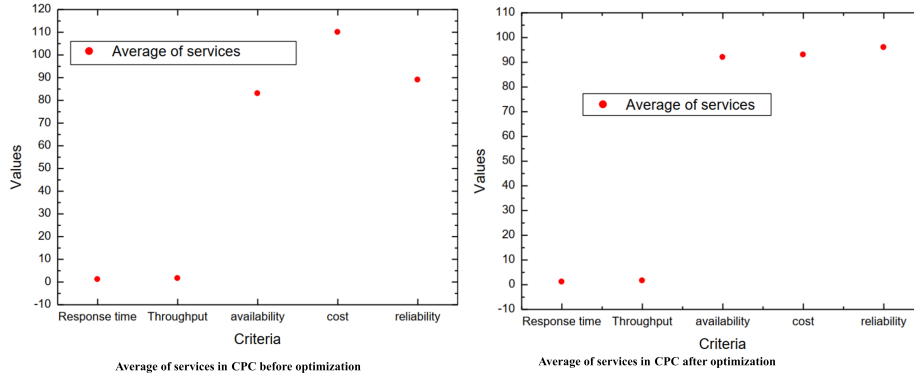


Figure 11. Average of services in CPC before and after optimization

6.2 Experiment 2

The goal of this experimentation is to show the average of services in the SPC before and after optimization. As shown in Figure 12, the values of the response time,

throughput, availability, reliability and cost before optimization were respectively 2.1, 2.74, 74 %, 82 % and $ 135. After optimization, the availability increases from 74 % to 89 %, reliability from 82 % to 93 % and the cost decreases from $ 135 to $ 100. In this experiment, we found that the number of crossing services led to an optimization in the values of three criteria.

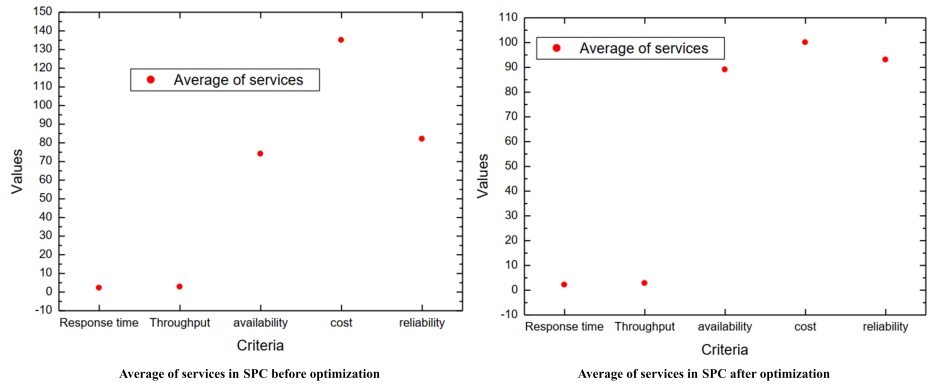


Figure 12. Average of services in SPC before and after optimization

6.3 Experiment 3

The goal of this experimentation is to show the average of services in the Common-Services-Class (CSC) before and after optimization. As shown in Figure 13, the values of the response time, throughput, availability, reliability and cost, before the optimization, are respectively 1.52, 2.35, 85 %, 81 % and $ 120. After the optimization, the availability increases from 85 % to 96 %, the reliability increases from 81 % to 94 % and the cost decreases from $ 120 to $ 93. Due to the number of intersecting services, we observe through this experiment that the values of the three criteria have been optimized.

6.4 Experiment 4

The goal of this experiment is to evaluate our system according to the number of services. To reach this goal, we vary the number of services from less than 100 to up to 1000, then observe the values of the criteria. The experimental results are shown in Figure 14. When the number of services is more than 100, the availability has increased from 75.35 % to 82 %, the reliability has increased from 80.52 % to 85 %, the cost has decreased from $ 141.58 to $ 130, and the response time and throughput are unchanged. When the number of services is more than 500 and less than 1000, the availability has increased from 75.35 % to 90 %, the reliability has increased from 80.52 % to 92 %, the cost has decreased from $ 141.58 to $ 115.4, and the response

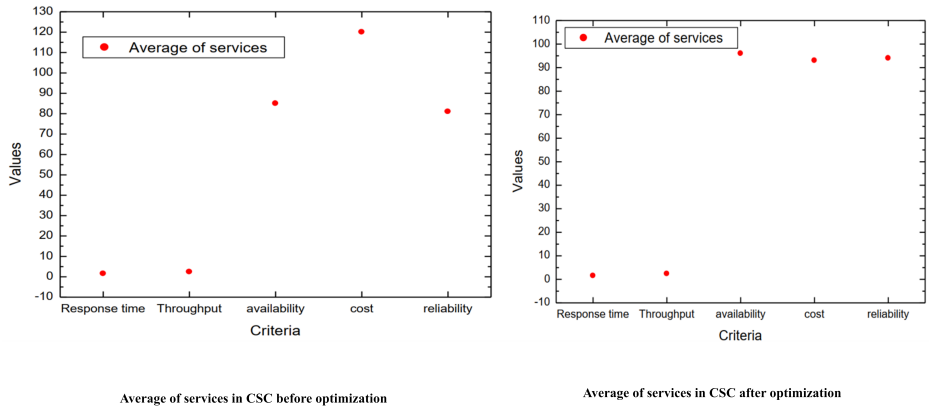


Figure 13. Average of services in CSC before and after optimization

time and throughput are unchanged. Finally, when the number of services is more than 1000, the availability has increased from 75.35 % to 99 %, the reliability has increased from 80.52 % to 98 %, the cost has decreased from $ 141.58 to $ 95, and response time and throughput are unchanged. From these results, we can conclude that the optimization rate increases accordingly with the increase of the number of services.

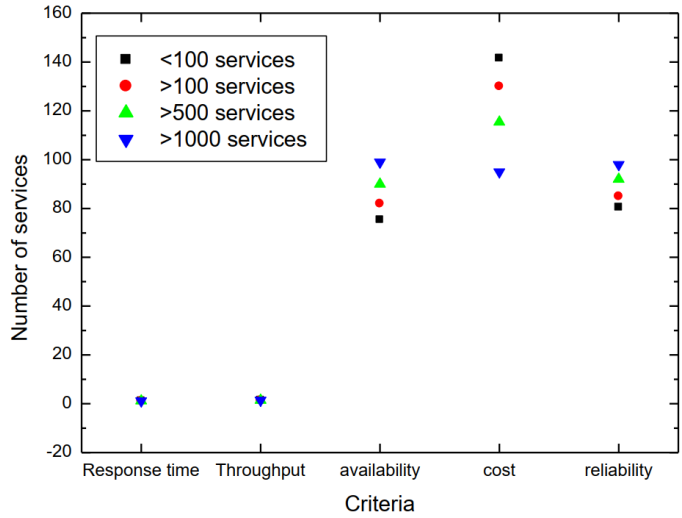


Figure 14. Impact of the services number on the optimization rate

7 CONCLUSION

Recently, with the development of cloud computing, the performance optimization of cloud services has become a very attractive research topic. This paper develops a new approach based on supervised learning and multi-criteria decision techniques to optimize cost and performance of services in a cloud-computing environment. Our approach uses an ANN to classify a set of cloud services. The inputs of this ANN are the service features, and its output is the classification results, consisting of two classes of cloud services. The first class comprises services preferred by the client, while the second class consists of services preferred by the providers. The ELECTRE method is employed to rank the services in both classes. Additionally, we made adaptations to the GA to suit our system. The GA produces a hybrid cloud service, which exists only in theory and not in practice. To resolve this issue, we used similarity tests to quantify the similarity level between the hybrid service and other advantages found in both classes. The experimental results demonstrate the effectiveness of MOOA-CSF. Nevertheless, certain limitations exist in this study, including the absence of optimization for response time and throughput. In our future research, we intend to expand our approach by incorporating other crucial criteria. Moreover, we aim to optimize response time and throughput using alternative optimization methods.

REFERENCES

[1] Ghobaei-Arani, M.—Rahmanian, A. A.—Souri, A.—Rahmani, A. M.: A Moth-Flame Optimization Algorithm for Web Service Composition in Cloud Computing: Simulation and Verification. Software: Practice and Experience, Vol. 48, 2018, No. 10, pp. 1865–1892, doi: 10.1002/spe.2598.

[2] Gayathri, V.—Selvi, S.—Kalaavathi, B.: Analysis on Cost and Performance Optimization in Cloud Scheduling. International Journal of Engineering Research and Technology (IJERT), Vol. 3, 2014, No. 11, pp. 929–934.

[3] Mishra, S. K.—Sahoo, B.—Parida, P. P.: Load Balancing in Cloud Computing: A Big Picture. Journal of King Saud University - Computer and Information Sciences, Vol. 32, 2020, No. 2, pp. 149–158, doi: 10.1016/j.jksuci.2018.01.003.

[4] Periasamy, R.: Performance Optimization in Cloud Computing Environment. 2012 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), 2012, pp. 1–6, doi: 10.1109/CCEM.2012.6354621.

[5] Sreenivasulu, G.—Paramasivam, I.: Hybrid Optimization Algorithm for Task Scheduling and Virtual Machine Allocation in Cloud Computing. Evolutionary Intelligence, Vol. 14, 2021, pp. 1015–1022, doi: 10.1007/s12065-020-00517-2.

[6] Abualigah, L.—Yousri, D.—Abd Elaziz, M.—Ewees, A. A.—Al-Qaness, M. A. A.—Gandomi, A. H.: Aquila Optimizer: A Novel Meta-Heuristic Optimization Algorithm. Computers and Industrial Engineering, Vol. 157, 2021, Art. No. 107250, doi: 10.1016/j.cie.2021.107250.

[7] Kaurav, N. S.—Yadav, P.: A Genetic Algorithm-Based Load Balancing Approach for Resource Optimization for Cloud Computing Environment. International Journal of Information and Computing Science, Vol. 6, 2019, No. 3, pp. 175–184.

[8] Zhou, Z.—Li, F.—Zhu, H.—Xie, H.—Abawajy, J. H.—Chowdhury, M. U.: An Improved Genetic Algorithm Using Greedy Strategy Toward Task Scheduling Optimization in Cloud Environments. Neural Computing and Applications, Vol. 32, 2020, pp. 1531–1541, doi: 10.1007/s00521-019-04119-7.

[9] Yiqiu, F.—Xia, X.—Junwei, G.: Cloud Computing Task Scheduling Algorithm Based on Improved Genetic Algorithm. 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), 2019, pp. 852–856, doi: 10.1109/ITNEC.2019.8728996.

[10] Shrimali, B.—Patel, H.: Multi-Objective Optimization Oriented Policy for Performance and Energy Efficient Resource Allocation in Cloud Environment. Journal of King Saud University - Computer and Information Sciences, Vol. 32, 2020, No. 7, pp. 860–869, doi: 10.1016/j.jksuci.2017.12.001.

[11] Bezza, Y.—Hioual, O.—Hioual, O.: A Multicriteria Decision Model for Optimizing Costs and Performances for a Cloud User. Distributed Sensing and Intelligent Systems: Proceedings of ICDSIS 2020, 2022, pp. 427–437, doi: 10.1007/978-3-030-64258-7_37.

[12] Ahmad, S. G.—Iqbal, T.—Munir, E. U.—Ramzan, N.: Cost Optimization in Cloud Environment Based on Task Deadline. Journal of Cloud Computing, Vol. 12, 2023, No. 1, Art. No. 9, doi: 10.1186/s13677-022-00370-x.

[13] Abdella, M.—Marwala, T.: The Use of Genetic Algorithms and Neural Networks to Approximate Missing Data in Database. IEEE 3rd International Conference on Computational Cybernetics (ICCC 2005), 2005, pp. 207–212, doi: 10.1109/ICCCYB.2005.1511574.

[14] Hicham, G. T.—Chaker, E. A.—Lotfi, E.: Comparative Study of Neural Networks Algorithms for Cloud Computing CPU Scheduling. International Journal of Electrical and Computer Engineering (IJECE), Vol. 7, 2017, No. 6, pp. 3570–3577, doi: 10.11591/ijece.v7i6.pp3570-3577.

[15] Aiello, G.—Enea, M.—Galante, G. M.: A Multi-Objective Approach to Facility Layout Problem by Genetic Search and Electre Method. Book Faim, 2005.

[16] Mary, S. A. S. A.—Suganya, G.: Multi-Criteria Decision Making Using ELECTRE. Circuits and Systems, Vol. 7, 2016, No. 6, pp. 1008–1020, doi: 10.4236/cs.2016.76085.

[17] Guo, L.—Yan, T.—Zhao, S.—Jiang, C.: Dynamic Performance Optimization for Cloud Computing Using M/M/M Queueing System. Journal of Applied Mathematics, Vol. 2014, 2014, Art. No. 756592, doi: 10.1155/2014/756592.

[18] Ismayilov, G.—Topcuoglu, H. R.: Neural Network Based Multi-Objective Evolutionary Algorithm for Dynamic Workflow Scheduling in Cloud Computing. Future Generation Computer Systems, Vol. 102, 2020, pp. 307–322, doi: 10.1016/j.future.2019.08.012.

[19] Simic, V.—Stojanovic, B.—Ivanovic, M.: Optimizing the Performance of Optimization in the Cloud Environment - An Intelligent Auto-Scaling Ap-

proach. Future Generation Computer Systems, Vol. 101, 2019, pp. 909–920, doi: 10.1016/j.future.2019.07.042.

[20] Vahidi, J.—Rahmati, M.: Optimization of Resource Allocation in Cloud Computing by Grasshopper Optimization Algorithm. 2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI), 2019, pp. 839–844, doi: 10.1109/KBEI.2019.8735098.

[21] Salem, R.—Salam, M. A.—Abdelkader, H.—Mohamed, A. A.: An Artificial Bee Colony Algorithm for Data Replication Optimization in Cloud Environments. IEEE Access, Vol. 8, 2019, pp. 51841–51852, doi: 10.1109/ACCESS.2019.2957436.

[22] Iwendi, C.—Maddikunta, P. K. R.—Gadekallu, T. R.—Lakshmanna, K.—Bashir, A. K.—Piran, M. J.: A Metaheuristic Optimization Approach for Energy Efficiency in the IoT Networks. Software: Practice and Experience, Vol. 51, 2021, No. 12, pp. 2558–2571, doi: 10.1002/spe.2797.

[23] Irshad, O.—Khan, M. U. G.—Iqbal, R.—Basheer, S.—Bashir, A. K.: Performance Optimization of IoT Based Biological Systems Using Deep Learning. Computer Communications, Vol. 155, 2020, pp. 24–31, doi: 10.1016/j.comcom.2020.02.059.

[24] Ivanovic, M.—Simic, V.—Stojanovic, B.—Kaplarevic-Malisic, A.—Marovic, B.: Elastic Grid Resource Provisioning with WoBinGO: A Parallel Framework for Genetic Algorithm Based Optimization. Future Generation Computer Systems, Vol. 42, 2015, pp. 44–54, doi: 10.1016/j.future.2014.09.004.

[25] Bansal, M.—Malik, S. K.: A Multi-Faceted Optimization Scheduling Framework Based on the Particle Swarm Optimization Algorithm in Cloud Computing. Sustainable Computing: Informatics and Systems, Vol. 28, 2020, Art. No. 100429, doi: 10.1016/j.suscom.2020.100429.

[26] Zhou, K.—Wen, Y.—Wu, W.—Ni, Z.—Jin, T.—Long, X.: Cloud Service Optimization Method Based on Dynamic Artificial Ant-Bee Colony Algorithm in Agricultural Equipment Manufacturing. Mathematical Problems in Engineering, Vol. 2020, 2020, pp. 1–11, doi: 10.1155/2020/9134695.

[27] Ragmani, A.—Elomri, A.—Abghour, N.—Moussaid, K.—Rida, M.: FACO: A Hybrid Fuzzy Ant Colony Optimization Algorithm for Virtual Machine Scheduling in High-Performance Cloud Computing. Journal of Ambient Intelligence and Humanized Computing, Vol. 11, 2020, pp. 3975–3987, doi: 10.1007/s12652-019-01631-5.

[28] Vinothini, C.—Balasubramanie, P.—Priya, J.: Hybrid of Meta Heuristic Firefly and Genetic Algorithm for Optimization Approach in the Cloud Environment. Webology, Vol. 17, 2020, No. 1, pp. 297–305, doi: 10.14704/WEB/V17I1/WEB17005.

[29] Lilhore, U. K.—Simaiya, S.—Maheshwari, S.—Manhar, A.—Kumar, S.: Cloud Performance Evaluation: Hybrid Load Balancing Model Based on Modified Particle Swarm Optimization and Improved Metaheuristic Firefly Algorithms. International Journal of Advanced Science and Technology, Vol. 29, 2020, No. 5, pp. 12315–12331.

[30] Abualigah, L.—Alkhrabsheh, M.: Amended Hybrid Multi-Verse Optimizer with Genetic Algorithm for Solving Task Scheduling Problem in Cloud Computing. The Journal of Supercomputing, Vol. 78, 2022, No. 1, pp. 740–765, doi:

10.1007/s11227-021-03915-0.

[31] Sharma, S.—Sharma, S.—Athaiya, A.: Activation Functions in Neural Networks. International Journal of Engineering Applied Sciences and Technology (IJEAST), Vol. 4, 2020, No. 12, pp. 310–316.

[32] Lin, L.—Cao, L.—Wang, J.—Zhang, C.: The Applications of Genetic Algorithms in Stock Market Data Mining Optimisation. Management Information Systems 2004, 2004, pp. 273–280.

[33] Biswas, P.—Pramanik, S.—Giri, B. C.: Cosine Similarity Measure Based Multi-Attribute Decisionmaking with Trapezoidal Fuzzy Neutrosophic Numbers. Neutrosophic Sets and Systems, Vol. 8, 2015, pp. 46–56.

[34] Al-Masri, E.—Mahmoud, Q. H.: Investigating Web Services on the World Wide Web. Proceedings of the 17th International Conference on World Wide Web, 2008, pp. 795–804, doi: 10.1145/1367497.1367605.

Youcef Bezza is a Ph.D. student in computer science, in security and web technology specialty at the Abbas Laghrour Khenchela University, Algeria. He obtained his Master's degree and Licence degree from the same university. His research interest includes cloud computing, optimization, IoT and fault tolerance.

Ouassila Hioual received her B.Sc. and M.Sc. in computer science from the Mentouri University of Constantine, Algeria in 2002 and 2005. She has worked as a Lecturer at the Department of Computer Science and Mathematics Science at the Mentouri University of Constantine, Algeria from 2005 to 2008. Currently, she works as Full Professor at the Department of Mathematics and Computer Science at Abbes Laghrour University of Khenchela, Algeria. She supervised many doctoral, master and licence students. Since September 2006 until October 2011, she prepared her Ph.D. in computer science. She has published a number of articles in international conferences and journals. Her research interests include CPS, Industry 4.0, data science, IoT and cloud computing environments, energy consumption.

Ouided Hioual received her B.Sc. degree in computer science from the Mentouri University of Constantine, Algeria in 2004, and M.Sc. degree in computer science from the Abbes Laghrour University of Khenchela, Algeria in 2009. Currently, she is working as Associate Professor at the Department of Mathematics and Computer Science at Abbes Laghrour University of Khenchela, Algeria. She has supervised many Ph.D., Master and Licence students since September 2012 until now. Since October 2010 until December 2019 she prepared her Ph.D. in computer science. She has published a number of articles in international conferences and journals. Her research interests include semantic web, ontology, web services, multiagent system, cloud computing, knowledge management, reasoning in artificial intelligence, engineering of human-machine interfaces and automatic treatment of natural language.

Derya Yiltas-Kaplan received her B.Sc., M.Sc., and Ph.D. degrees in computer engineering from the Istanbul University, Istanbul, Türkiye, in 2001, 2003, and 2007, respectively. She completed her postdoctoral research with the North Carolina State University. She is currently Associate Professor with the Department of Computer Engineering, Istanbul University-Cerrahpaşa. Specializing in computer science, her research is notably focused on computer networks and data routing, making significant contributions to engineering and technology. She received a Postdoctoral Research Scholarship from The Scientific and Technological Research Council of Türkiye (TUBITAK).

Zeynep Gürkaş-Aydin completed her undergraduate studies in computer engineering at the Istanbul University Engineering Faculty in 2003. She completed her Master's degree in computer engineering at Istanbul University Institute of Science in 2005, her first Ph.D. in computer engineering at the Istanbul University Institute of Science in 2011, and her second Ph.D. in the field of informatics as part of the Ecole Doctorale program at Université Pierre-et-Marie-Curie: Paris VI in France in 2014. Currently, she serves as a faculty member in the Department of Cybersecurity at Istanbul University-Cerrahpaşa Engineering Faculty. Her research covers a wide range of topics in computer science and engineering.

Computing and Informatics, Vol. 42, 2023, 923–942, doi: 10.31577/cai_2023_4_923

RETRIEVAL TECHNOLOGY OF ENTERPRISE DATA CENTER RESOURCES BASED ON DENSITY PEAK CLUSTERING ALGORITHM

Jiaming Jiang*, Guoheng Ruan, Zhenggan Dai

Qingyuan Power Supply Bureau of Guangdong Power Grid Co., Ltd.
Qingyuan, 511500, China
e-mail: {jiangjiaming0620, ruanguoheng, daizenggan2022}@163.com

Abstract. In order to effectively ensure the retrieval effect of enterprise data center resources, improve the retrieval accuracy of enterprise data center resources, and shorten the retrieval time of enterprise data center resources, a retrieval technology of enterprise data center resources based on density peak clustering algorithm is proposed. Analytical clustering algorithms, density clustering algorithms, and density peak clustering algorithms are all types of clustering algorithms. To reduce the dimensionality of enterprise data center resources, the kernel principal component analysis method is used. The structure of the enterprise data center resource set is reorganized and the feature quantity of the enterprise data center resource distribution is extracted using feature space reorganization technology. On this basis, the density peak clustering is carried out on the data center resource set of enterprise, and the semantic association distribution model of data center resource retrieval in enterprise is constructed. Through the semantic registration and weighted vector combination control method, the retrieval of enterprise data center resources is realized. The experimental results show that the proposed algorithm has a good effect on the retrieval of enterprise data center resources, which can effectively improve the resource retrieval accuracy and shorten the resource retrieval time.

Keywords: Density peak clustering algorithm, enterprise data center, kernel principal component analysis method, resource retrieval, semantic correlation distribution

* Corresponding author

1 INTRODUCTION

The data center is the precipitation of the existing/new information system business and data, and is the middle and supporting platform for realizing data empowerment of new business and new applications [1, 2, 3]. Enterprises build a data center, and through the aggregation and reuse of business and data, guide the company's power grid, industry, finance and international sector resources, systems and data integration. Effectively improve the rapid response and flexible adjustment capabilities of the information system, effectively empower front-end business applications and enhance innovation capabilities. As an important platform for data sharing and analysis within the enterprise, the enterprise data center carries the massive data resources of the enterprise, and at the same time realizes rapid response support for various data application needs. With the advancement of the construction of the enterprise data center, the demand for front-end business applications will also increase rapidly. Faced with more and more data resources and data services, how to quickly support business applications and help users find, see, and understand data in an extremely fast way is particularly important. Therefore, it is necessary to apply data resource retrieval technology to quickly and accurately find the desired content from massive information [4, 5, 6]. The pros and cons of the data resource retrieval technology largely determines the utilization rate of the data resources in the data center, which makes the data resource retrieval technology more important.

At present, scholars in related fields have carried out research on data resource retrieval, and have made great progress. Reference [6] proposes a retrieval algorithm for online tourism resources based on Page Rank search and ranking algorithm. A topic collection algorithm is constructed, and a starting point, topic keywords, and prediction mechanism are established. The algorithm consists of three stages: the first climbing stage, the learning stage and the continuous climbing stage. Open directory search was used for similarity judgment and result evaluation. Wood extraction algorithm is based on network tourism resource density. The algorithm calculates the proportion of Internet tourism resource labels by row, and uses a threshold extraction algorithm to distinguish regions from private non-Internet tourism resource regions. This paper takes tourism network resource monitoring as the research object, and establishes a tourism network resource monitoring system, which can provide users with customizable, all-round, real-time tourism network resource collection, extraction and retrieval services, so as to monitor tourism resources. This method can successfully extract the main content of articles from various web pages. The research results of this paper can promote the construction of tourism informatization, help users master the latest tourism information, and bring great convenience to the tourism industry. The system only downloads tourism-related information through theme collection technology, reducing the interference of irrelevant redundant web pages. Reference [7] proposed an image network teaching resource retrieval algorithm based on deep hashing algorithm. A pixel big data detection model of multi-view attribute coding image network teaching resources

is constructed, and the pixel information collected by multi-view attribute coding image network teaching resources is reconstructed. The fuzzy information feature components of the multi-view attribute-encoded images are extracted, and the edge contour distribution images are combined. Distributed fusion network teaching resource view image edge contour, realizes the construction of view feature parameter set. The gray invariant moment feature analysis method is used to complete the information encoding, and the deep hash algorithm is used to realize the retrieval of multi-view attribute-encoded image network teaching resources. The algorithm has a high level of resource fusion for multi-view coded image network teaching resource retrieval. However, the above methods still have the problems of poor resource retrieval effect, low precision and long time.

Aiming at the above problems, this paper proposes a data center resource retrieval technology for enterprise based on density peak clustering algorithm. The kernel principal component analysis method is used to reduce the dimensionality and process of enterprise data center resources. Combined with feature space reorganization technology, the feature quantity of resource distribution in enterprise data center is extracted. On this basis, the density peak clustering is carried out on the enterprise data center resource set, and the enterprise data center resource retrieval is realized through semantic registration and weighted vector combination control method. The resource retrieval effect of the algorithm is good, which can effectively improve the resource retrieval accuracy and shorten the resource retrieval time.

2 RELEVANT BASIC THEORIES

2.1 Clustering Algorithm

Clustering is an important data mining technique. Clustering can find hidden patterns and trends in data without any supervised information such as data labels [8]. Graph analytics is considered as most influential tool able to guide to unwrap the hidden patterns and relationships in the data. Therefore, from a machine learning point of view, clustering is a form of unsupervised learning, where the cluster classes correspond to latent structures. Here, a simple definition can be presented: given a set of data points, they are divided into multiple clusters, where similar data points are in the same cluster, and dissimilar data points are in different clusters.

A clustering algorithm usually involves four steps: data representation, modeling, optimization, and validation. The data representation predetermines which data type is used to analyze the data. Data representation defines the form in which data is stored, processed and transmitted. On the basis of data representation, the modeling phase defines the concept of cluster class, that is, the data objects are divided. Typically, the quality of this partition is assessed by an approximate metric. For clustering analysis, the Gaussian mixture of models is considered as the most prominent model which says that the dataset is generally modelled with fixed number of Gaussian supplies. The optimization stage is to optimize or approx-

imate the above-mentioned quality evaluation criteria when finding these hidden cluster structures according to the clustering goal. This mass is optimized or approximately optimized to produce a clustering solution. The Elbow method is the common method for defining the optimal number of clusters. The verification stage is to evaluate the clustering results obtained through the above scheme.

Let a dataset contain Q data objects, which can also be called data records or data points, expressed as:

$$W = \{w_1, w_2, \ldots, w_Q\}. \tag{1}$$

Each data object can be represented by an E-dimensional vector as:

$$w_i = (w_{i,1}, w_{i,2}, \ldots, w_{i,E})^T. \tag{2}$$

In Equation (2), $w_{i,E}$ is the E attribute of w_i, which can also be called a feature or dimension. The number of attributes E is also known as the dimension of the dataset. The dataset can be divided into clusters and expressed as:

$$R = \{R_1, R_2, \ldots, R_T\}. \tag{3}$$

In Equation (3), Y is the number of clusters, and the division satisfies the following conditions:

$$W = R_1 \cup \cdots \cup R_Y \cup R_O. \tag{4}$$

And for all $i, E = 1, 2, \ldots, Y, i \neq E$, hard clustering is expressed

$$R_i \cap R_E = \emptyset. \tag{5}$$

Hard clustering is a process of grouping the data such that an item can exist in various clusters. This method is also known as non-fuzzy clustering. The data points usually belong to the various clusters in hard clustering.

Since clustering is a fairly common problem, clustering algorithms can be divided into several categories based on clustering techniques: Model-based clustering algorithms, distance-based clustering algorithms, density and grid-based clustering algorithms, graph-based clustering, subspace clustering, etc. The classification of clustering algorithms is shown in Figure 1.

1. Model-based clustering is based on the assumption: the data is generated by a mixture of multiple probability distributions. Probability distributions are utilized to explain the real-life variables populations and it is also used in hypothesis testing for determining values of p. It is commonly considered as the statistical function which defines all the conceivable values. Therefore, this type of clustering method estimates the parameters of each distribution in order to perfectly fit the observed data. The whole clustering process is to first assume a specific generative model, and then use the expectation maximization algorithm (EM, Expectation Maximization) to estimate the parameters of

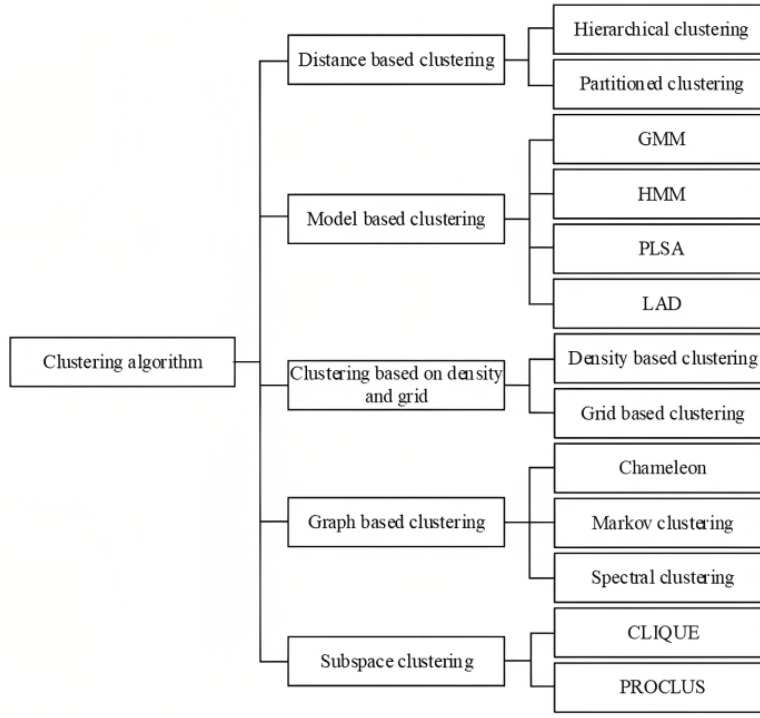


Figure 1. Classification of clustering algorithms

the model. Expectation Maximization is an estimation of mixture model which is the technique of probability density used in several applications. For implementing the distribution-based algorithm, the Data mining used the expectation maximization.

2. Distance-based clustering method can be regarded as a special form based on model. For example, the k-Means algorithm is closely related to the Gaussian distribution. Because, distance-based clustering methods are easy to implement and very simple in various scenarios such as consumer datasets, statistical profiles (age, gender, income, etc.), purchase history and web browsing activities. In order to understand a broad group of customers' behavior and preferences, such data are utilized to identify them together with their attributes. Therefore, such clustering algorithms are widely used. Distance-based clustering algorithms can generally be divided into two categories: hierarchical clustering algorithms and partition clustering algorithms.

   Hierarchical clustering algorithms: An algorithm is the analysis of hierarchical cluster which combines the comparable object into groupings called as clusters. A series of partitions of data objects has been developed by this method.

Partition clustering algorithm: This kind of algorithm is generally converting a distribution of set of data objects into form of clusters and so that every data object is in the form of one subset. A partitional clustering can be able to decay the set of data into set of disjoint clusters.

3. Density- and Grid-based clustering are two closely related clustering methods. Both types of approaches attempt to explore the data space at a high granularity level.

   Density-based clustering: Density-based clustering methods are built on a fundamental assumption: clusters are defined as dense regions separated by low-density regions. In k-means algorithm, the distance between the points has been determined by using distance calculation technique. And the Euclidean distance method is considered as the most frequently used method. By applying such phases, the density-based algorithm can be able to discover high density regions and disperse them from low density regions. It partitions the data in the data space at a higher granularity and "merges" dense regions together. At a high level of granularity, the solutions of density and grid-based clustering are the major two closely correlated categories which attempt two categories attempt to discover the data space. The density and grid-based cluster allows the arbitrary shape cluster and categorize the outliers in the data.

   Grid-based clustering: Grid-based clustering methods are a special type of density-based clustering methods in which the explored data space is partitioned into a grid-like structure.

4. Graph-based clustering is a class of clustering algorithms based on graph theory. The main idea of this type of clustering is to regard data objects as complete graphs or nodes in connected graphs and define the weight between nodes as the distance between data objects. In the simplest way to describe, all nodes form a complete graph connected to each other, using the similarity between data objects to represent the edges of the graph, then the data can be represented in the form of a weighted complete graph. When similar data samples are gathered into only one cluster, the data objects are frequently used in clustering method. Other data samples are gathered into various ones. After the graph is created, some of the edges are removed by some rules. Based on the weights of the cluster all the edges of the graphs are categorized in the form of descending order. And so, for every edge, starting from the maximum weighted edge, all the edge has been removed when the weight of the current edge is much greater than the number or the sum of adjacent edges. Further, remove each edge that has a weight greater than the average adjacent edge weight. After the above process, a disconnected graph is obtained, in which each subgraph represents a cluster class.

5. Subspace clustering methods can be considered as a form of local feature selection or local dimensionality reduction, which performs feature selection or

transformation for different subsets of the data. The techniques of dimensionality reduction might be well-defined as it was path of changing the higher dimensions dataset into lower dimension dataset which also ensured that it delivers alike information's. when resolving the classification, these methods are broadly utilized in machine learning for gaining good fit predictive model. Subspace clustering can be seen as an extension of traditional clustering. The subspace clustering algorithm is a kind of algorithm which is able to locate clusters within various subspaces. Generally speaking, subspace clustering can be divided into two categories: Bottom-up algorithms and Top-down algorithms.

2.2 Density Clustering Algorithm

Density-based clustering is a class of nonparametric methods that treat regions of high density as clusters [9]. Density-based clustering methods are based on the assumption of density-based clusters. A density-based cluster is a contiguous set of high-density regions separated by low-density regions. The density-based clustering method is one of the most famous unsupervised learning methods which is used in the algorithm of machine learning and model building. By the two low point density clusters, the data points have been separated and such clusters are regarded as noise.

This class of methods relies on two important parameters: the distance threshold and the density threshold. The distance threshold and the density threshold parameters play vital roles in the density-based clustering algorithm. Further, the distance threshold defines that the separation between two major consecutive elements in the cluster is curved to the following decimal point whereas the density threshold has been well-defined by two parameters such as the neighborhood radius, i.e. eps, and data points numbers, i.e. minPts. These two can be comprehended if prospecting the two concepts called density reachability and connectivity. One of the most important concepts in density-based methods is the nearest neighbor of a data point, which is defined as follows: the nearest neighbor is considered the most essential in cluster algorithms. Also, the analysis of the nearest neighbor is expressive statistics which displays locating feature patterns by contrasting the clearly observed nearest neighbor distance. It is also considered one of the simplest procedures which can be able to classify the rule of the nearest neighbor.

The α nearest neighbor of a data point $u \in U$ is:

$$Q_\alpha(u) = \{o \in U : \mathrm{dist}(u, o) \leq \alpha\}. \tag{6}$$

In Equation (6), $dist(,)$ is a specific distance function.

1. Directly density-reachable: If a data point $u \in U$ satisfies $u \in Q_\alpha(o)$ and $|Q_\alpha(o)| \geq p$, then the data point u is directly density-reachable from the data point o.
2. Density-reachable: If there is a sequence of data points denoted by $\{u_{a1}, u_{a2}, \ldots, u_{an}\}$, $u_{a1} = u$, $u_{an} = 0$, such that for $i = 1, 2, \ldots, n-1$, there is a direct density

reachable from data point u_{an+1} to u_{an}, and data point u from data point o is density reachable.

3. Density-connected: If there is a data point with the given parameters α and p and data points are density reachable from the data point, the two data points are then density-connected using these parameters.

A cluster class can be defined as the largest set of densely connected points. Mathematically, it can be defined as follows: Cluster: Given parameters α and p, a cluster needs to satisfy the following two properties:

1. Maximality: For $\forall u, o$, if $u \in C$ is density-reachable from u to o, then $o \in C$.
2. Connectivity: For $\forall u, o \in C, u$ and o are densely connected.

Based on the two parameters α and p and the above definitions, density-based clustering can identify three different types of data points:

1. Core point: A core point is a data point with high-density neighbors, that is, the number of α nearest neighbors is greater than or equal to the density threshold p.
2. Border point: The border point also belongs to a certain cluster, but its neighbors are not dense. According to the border point of the clusters, the latent cross cluster is utilized for removing the edges of the cross-cluster.
3. Noise point: The noise point does not belong to any cluster whereas the noise point has been considered as neither a core point nor a border point. Also, it classifies arbitrary size clusters in the database along with outliers. A noise point is not assigned for two points such as core and border points.

For data points that are both core points, the direct density reachability is a symmetric relationship, but if a core point and a boundary point are involved, the direct density reachability is usually not symmetrical. Density accessibility, as an extension of direct density accessibility, is also asymmetric.

2.3 Density Peak Clustering Algorithm

The density peak clustering algorithm is a new clustering algorithm that can find non-convex clusters. Density peak cluster classify the centers of cluster at ease without any preceding data. The non-convex clusters are the clusters which are usually incapable to recognize the clusters along with the shapes of non-convex specially the manifold ones. The core idea of the algorithm is based on two important assumptions about the cluster center point or the density peak point [10, 11].

Assumption 1. The local density of the cluster center point is greater than the local density of its surrounding neighbors.

Assumption 2. There is a relatively large distance between the cluster center point and other center points.

The above two assumptions not only describe the cluster center point but also give a criterion for detecting the center point. To quantify the above assumptions, two important values are introduced for each data point u_i: the distance δ_i of the data point whose local density χ_i is greater than it and is closest to it.

The local density χ_i is defined as:

$$\begin{aligned} \chi_i &= \sum_{u_j \in U} \varepsilon \left(\operatorname{dist} \left(u_i, u_j \right) - d_c \right), \\ \varepsilon(u) &= \begin{cases} 1, & u < 0, \\ 0, & u \geq 0. \end{cases} \end{aligned} \tag{7}$$

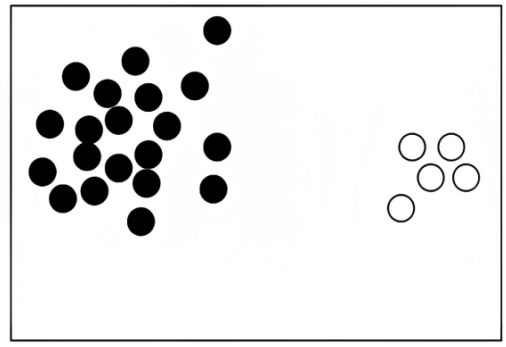
In Equation (7), d_c is called the cutoff distance, which is the only input parameter of the density peak clustering algorithm and actually plays the role of the distance threshold.

The data point distance δ_i is defined as:

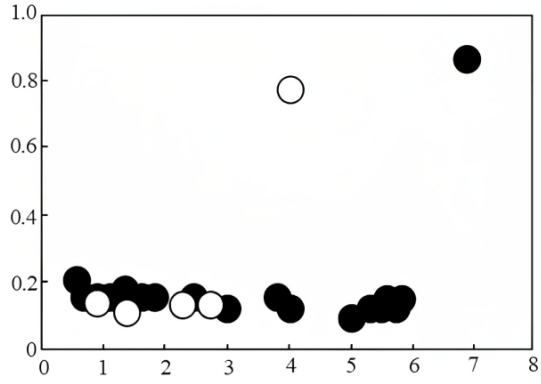
$$\delta_i = \begin{cases} \min_{j:\chi_i < \chi_j} \left(\operatorname{dist} \left(u_i, u_j \right) \right), & \text{if Jjs.t.} \chi_i < \chi_j, \\ \min_j \left(\operatorname{dist} \left(u_i, u_j \right) \right), & \text{otherwise.} \end{cases} \tag{8}$$

Similar to the clustering algorithm based on the center point, the density peak clustering algorithm also needs to find the center point (density peak) of the cluster. In order to obtain the cluster center point, the density peak clustering algorithm constructs the decision graph of χ and δ, and selects the data point with larger χ and δ as the cluster center. Simply put, the data point at the top right of the decision diagram is manually selected as the cluster center. The 2D medium density peak clustering algorithm is shown in Figure 2.

In Figure 2, 25 data points are shown, and the decision graph for this data contains χ and δ for each point. Figure 2 b) is above the density peak points, these two points have high δ and relatively high χ. However, the distribution of the remaining points is different from the clustering algorithm based on the center points. In terms of allocation strategy, this algorithm is similar to density-based clustering algorithm or agglomerative hierarchical clustering algorithm, and the attribution of data points depends on the attribution of surrounding points. The agglomerative clustering is defined as most common type of hierarchical clustering which is used for collection of objects in cluster based. It is also known as AGNES which defines Agglomerative Nesting. By treating every object as a single cluster, this algorithm has been initiated. For the remaining data points u_i, the density peak clustering algorithm classifies them into the cluster class of the data points whose density is greater than u_i and is the closest to u_i, and only needs to complete the assignment of the remaining data points in one step.



a) Data point distribution



b) Decision start

Figure 2. 2-dimensional medium density peak clustering algorithm

3 RETRIEVAL ALGORITHM OF ENTERPRISE DATA CENTER RESOURCE

3.1 Dimensionality Reduction Processing Enterprise Data Center Resources

Because there are many kinds of characteristic data in the enterprise data center resources, the distribution of data points in space is relatively sparse, and the similarity measurement method in low-dimensional space cannot effectively process high-dimensional data. The similarity measurement method measures data mining building blocks and utilized in recommendation engines, methods of clustering and detecting anomalies. This method is a distance along with dimensions to express the object features. The similarity between two data objects is determined by using the measurements method. To this end, the kernel principal component analysis method [12, 13] is used to reduce the dimensionality of enterprise data center re-

sources. In kernel principal component analysis, for the reduction of dimensionality, the linear, polynomial, sigmoid kernel methods are used. As the reduction of dimensionality k-means clustering is put in to the document's reduction. Assuming that the resource set in the original enterprise data center is $D = \{d_1, d_2, \ldots, d_n\} \in R^m$, the nonlinear mapping is:

$$\begin{aligned} &\phi : R^m \to F, \\ &d \mapsto g = \phi(d), \end{aligned} \tag{9}$$

Centralize the enterprise data center resources mapped in the high-dimensional space, and then solve the covariance matrix of the data center resources of the high-dimensional enterprise:

$$H_J = \frac{1}{n-1} \sum_{i=1}^{n} \phi\left(d_i\right) \phi\left(d_i\right)^T = \frac{1}{n-1} \sum_{i=1}^{n} g_i g_i^T. \tag{10}$$

Eigendecompose H_J to solve the eigenvector matrix:

$$H_J h_r = \varphi_r h_r. \tag{11}$$

But:

$$g_i^T H_J h_r = \varphi_r g_i^T h_r. \tag{12}$$

Substitute Equation (10) and Equation (11) into Equation (12) to get:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \gamma_k^r \sum_{k=1}^{n} \left(g_i^T g_j\right) \left(g_i^T g_j\right) = \sum_{i=1}^{n} (n-1) \varphi_r \sum_{k=1}^{n} \gamma_k^r \left(g_i^T g_k\right). \tag{13}$$

Introduce the kernel function:

$$K_{jj} = {g_i}^T g_j. \tag{14}$$

Substitute Equation (14) into Equation (13), and simplify to get:

$$K\gamma^r = \varphi_r \gamma^r. \tag{15}$$

Using the Equation (15) to eigendecompose the kernel function matrix K, γ^r can be obtained, and further eigenvectors can be obtained. Use the obtained feature vector to perform feature transformation on the high-dimensional enterprise data center resource set:

$$g = \eta^T g_i = \eta^T \phi\left(d_i\right) = \sum_{k=1}^{n} \gamma_k K\left(d_k, d_i\right). \tag{16}$$

When non-linear mapping is performed on the resource set in the original enterprise data center, the specific form of the non-linear mapping is not clear. Therefore, by selecting an appropriate kernel function K, performing eigendecomposition, and

obtaining the eigenvector γ^r, further, the resource g_i of the enterprise data center after dimensionality reduction can be obtained.

3.2 Extracting the Distribution Characteristics of Enterprise Data Center Resources

After dimensionality reduction and processing of enterprise data center resources, combined with feature space reorganization technology, the enterprise data center resource set structure reorganization is carried out, and the feature quantity of enterprise data center resource distribution is extracted. Calculate all link gain values and use the deep learning method [14] to obtain the clustering similarity feature of enterprise data center resources as:

$$K_L = \frac{\lambda\left(g_i + b\right)}{z}. \tag{17}$$

In Equation (17), λ represents the value result of enterprise data center resources defined by the standard. If the link gain value is $z \geq 1$, the fitness weight coefficient of enterprise data center resources is obtained as a positive number. Modify the clustering parameters of the resource set in each enterprise data center, and obtain the clustering effectiveness evaluation parameter distribution set and index weight of all cluster head nodes. A cluster head is considered as node which collects information's from the cluster sensors and sending information's to the base stations. The cluster heads are the nodes that pretend as a head of cluster. Based on the above analysis, a deep learning model for clustering enterprise data center resource sets is established, and the fuzzy feature distribution of enterprise data center resource sets is obtained, and the constraint programming model of enterprise data center resource sets is obtained as follows:

$$\min(F) = \sum_{i=1}^{m}\sum_{j=1}^{n} g_{ij}K_{ij}. \tag{18}$$

For the best cluster centers of all cluster head nodes, through ambiguity detection, the evaluation set and test set of enterprise data center resource set are obtained. After suspending the transmission of enterprise data center resources, the characteristic distribution of the enterprise data center resource set is obtained as follows:

$$D_q = \kappa \times \mu^2 + \upsilon \times \mu. \tag{19}$$

In Equation (19), κ represents the fused clustering feature set of enterprise data center resources, μ represents the pixel brightness value before correction of the fused clustering feature set of the resource set in the enterprise data center, and υ is the regression parameter. According to the extraction results of the distribution characteristics of the resources in the above enterprise data center, the density peak fusion clustering is performed.

3.3 Peak Clustering of Set Density in Enterprise Data Center Resources

On the basis of extracting the distribution characteristics of enterprise data center resources, the density peak clustering is performed on the resource sets of the enterprise data center. According to the grid block distribution of the resource set in the enterprise data center, the density peak feature quantity of the resource set in the enterprise data center is extracted, and the iterative formula of the algorithm for extracting the density peak feature is given as follows:

$$\varpi(\theta) = D_q \times \varpi(\theta - 1). \tag{20}$$

In Equation (20), ϖ represents the choice to adopt the embedded dimension scheduling value. Assuming that θ is the boundary value vector of the resource set in the enterprise data center transmitted by the cluster head node, the kurtosis of the resource set in the enterprise data center is defined as:

$$\theta_{kurt}(v) = \varpi(\theta) + E\left(v_1 + v_2\right) + \vartheta. \tag{21}$$

In Equation (21), v_1 and v_2 represent the boundary feature quantities of the resources in the enterprise data center, respectively, and ϑ is represented as a scalar. Through density peak clustering, the density distribution of resources in enterprise data center can be obtained to satisfy the following formula:

$$S_{xy} = \left\|B_x - B_y\right\|^2. \tag{22}$$

In Equation (22), x and y respectively represent any two nodes in the resource density distribution of enterprise data center resources, and B_x and B_y respectively represent the corresponding density pixel values of the two.

3.4 Realization of Resource Retrieval in Enterprise Data Center

Based on the clustering results of the peak density of resource sets in enterprise data center, a semantic correlation distribution model for resource retrieval in enterprise data center is constructed. Through the semantic registration and weighted vector combination control method [15], the retrieval of enterprise data center resources is realized. Using the joint quantitative feature analysis method, the least squares fitting function for the retrieval of enterprise data center resources is obtained [16]. The least square method finds a regression line or best-fitted line for any data set which is labelled by an equation. The core functions of the least square method reduce the sum of the squared errors. The description is as follows:

$$\mid \rho(x) = \varpi\left(S_{xy}\right)\sigma_{xy}. \tag{23}$$

In Equation (23), $\sigma_{xy} = 1$ indicates that the output of resource retrieval in enterprise data center satisfies convergence, and $\sigma_{xy} = 0$ indicates that the output

of resource retrieval in enterprise data center diverges. Therefore, the retrieval of enterprise data center resources is constructed, and the entropy function of the information distribution of enterprise data center resources is as follows:

$$\zeta(x) = \frac{\rho(x) - \vartheta}{k+1} + (1 - \theta_{\text{tot}}(v))\,\tau(k). \tag{24}$$

In Equation (24), $\tau(k)$ is the characteristic function. According to the entropy distribution, combined with the mean function detection method, the retrieval of resources in the enterprise data center is carried out, and the following results are obtained:

$$v(x) = g(x)(k+1) + e(t) + \xi\zeta. \tag{25}$$

In Equation (25), $e(t)$ is the set of attribute names, and ξ and ζ are the closeness functions of the distribution of enterprise data center resources. According to the above analysis, the retrieval of enterprise data center resources is realized. The algorithm implementation process is shown in Figure 3.

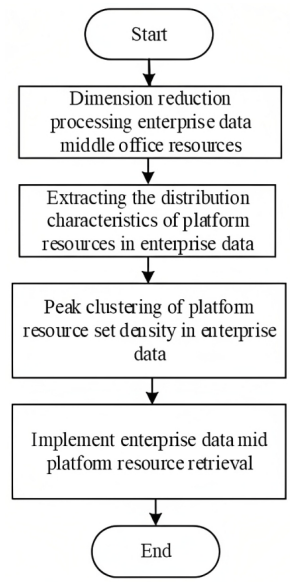


Figure 3. Algorithm implementation flow chart

4 EXPERIMENTAL SIMULATION AND ANALYSIS

4.1 Setting Up the Experimental Environment

In order to verify the effectiveness of the resource retrieval technology in enterprise data center based on the density peak clustering algorithm, the operating system used in the experiment is Windows 10, and the integrated development environment is MATLAB 2014a. The hardware conditions are: CPU Intel (R) Core (TM) i7-7700, main frequency 3.6 GHz, memory 8 GB. Selecting 5 000 MB enterprise data center resources as the experimental data, the proposed algorithm, the algorithm of reference [6] and the algorithm of reference [7] are compared to verify the effectiveness of the proposed algorithm.

4.2 Comparative Analysis of Retrieval Effect of Enterprise Data Center Resources

In order to verify the retrieval effect of the proposed algorithm in enterprise data center resources, the retrieval coverage is taken as the evaluation index. The higher the retrieval coverage, the better the retrieval effect of the algorithm's enterprise data center resources. The algorithm of reference [6], the algorithm of reference [7] and the proposed algorithm are used to compare, and the comparison results of the retrieval coverage of enterprise data center in different algorithms are shown in Figure 4.

Analysis of Figure 4 shows that when the amount of enterprise data center resources is 5 000 MB, the average retrieval coverage of enterprise data center resources of the algorithm of reference [6] is 84.6 %. The average enterprise data center resources retrieval coverage rate of the algorithm of the reference [7] is 79.7 %. The average enterprise data center resources retrieval coverage rate of the proposed algorithm is as high as 98.2 %. From this, it can be seen that the retrieval coverage of enterprise data center resources in the proposed algorithm is relatively high, indicating that the proposed algorithm has a better retrieval effect on the retrieval of enterprise data center resources.

4.3 Comparative Analysis of Retrieval Accuracy of Enterprise Data Center Resources

In order to further verify the retrieval accuracy of the proposed algorithm in enterprise data center resources, the retrieval accuracy rate is used as the evaluation index. The higher the retrieval accuracy is, the higher the retrieval accuracy of the algorithm's enterprise data center resources is. The algorithm of reference [6], the algorithm of reference [7] and the proposed algorithm are used to compare, and the comparison results of the retrieval accuracy of enterprise data center resources of different algorithms are shown in Figure 5.

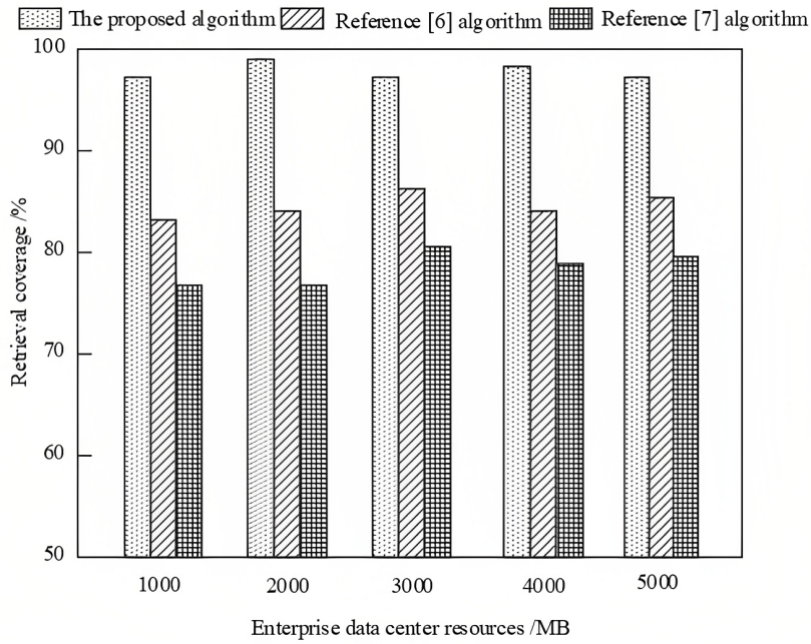


Figure 4. Comparison results of the retrieval coverage ratio of enterprise data center resources of different algorithms

Analysis of Figure 5 shows that when the amount of enterprise data center resources is 5 000 MB, the average retrieval accuracy rate of enterprise data center resources of the algorithm of reference [6] is 89.1 %. The average retrieval accuracy rate of data center resources in the algorithm of reference [7] is 86.4 %. And the average enterprise data center resources retrieval accuracy rate of the proposed algorithm is as high as 98.9 %. It can be seen that the proposed algorithm has a higher accuracy rate of retrieval of enterprise data center resources, indicating that the retrieval accuracy of the proposed algorithm in enterprise data center resources is higher.

4.4 Comparative Analysis of Retrieval Time of Enterprise Data Center Resources

On this basis, verify the retrieval time of the proposed algorithm in enterprise data center resources. The proposed algorithm, the algorithm of reference [6] and the algorithm of reference [7] are used to compare, and the comparison results of the retrieval time of enterprise data center resources of different algorithms are shown in Table 1.

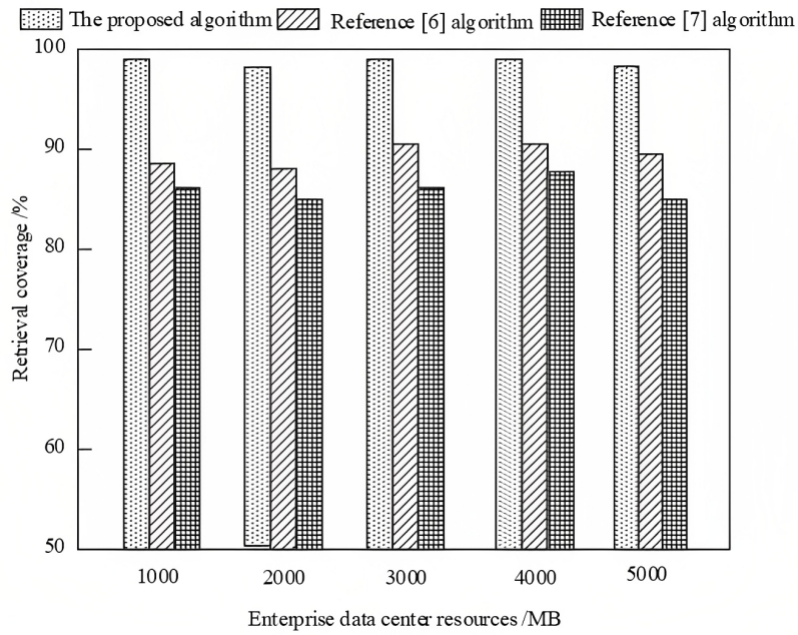


Figure 5. Corresponding author Comparison results of the accuracy rate of resource retrieval in enterprise data center in different algorithms

From the analysis of Table 1, it can be seen that with the increase of the amount of resources in the enterprise data center, the retrieval time of the enterprise data center resources of different algorithms increases accordingly. When the amount of enterprise data center resources is 5 000 MB, the retrieval time of enterprise data center resources in the algorithm of reference [6] is 12.3 s. The retrieval time of enterprise data center resources of the algorithm of reference [7] is 15.8 s. However, the retrieval time of the proposed algorithm in enterprise data center resources is

Enterprise Data Center Resources [MB]	The Proposed Algorithm [s]	The Algorithm of Reference [6] [s]	The Algorithm of Reference [7] [s]
1 000	1.2	4.5	7.2
2 000	2.6	6.1	9.8
3 000	4.1	8.8	11.6
4 000	5.2	10.6	13.2
5 000	6.7	12.3	15.8

Table 1. Comparison results of retrieval time of enterprise data center resources of different algorithms

only 6.7 s. It can be seen that the retrieval time of enterprise data center resources of the proposed algorithm is shorter.

5 CONCLUSION

In this paper, the retrieval technology of enterprise data center resources based on the density peak clustering algorithm is proposed, and the density peak clustering algorithm is used to realize enterprise data center resource retrieval. The algorithm has better retrieval effect of enterprise data center resources, which can effectively improve the retrieval accuracy of enterprise data center resources and shorten the retrieval time of enterprise data center resources. But the time complexity of this algorithm is high. Therefore, in the following research, how to reduce the time complexity of this process is the focus of the research.

Acknowledgments

Science and technology project of the China Southern Power Grid Corporation (Grant No. 031800kk52200001 (gdkjxm20200339)).

REFERENCES

[1] Zhou, A.—Wang, S.—Hsu, C. H.—Kim, M. H.—Wong, K. S.: Virtual Machine Placement with (m, N)-Fault Tolerance in Cloud Data Center. Cluster Computing, Vol. 22, 2019, No. 5, pp. 11619–11631, doi: 10.1007/s10586-017-1426-y.

[2] Mirza, J.—Raza, A.—Atieh, A.—Iqbal, S.—Ghafoor, S.: Self Restorable Intra Data Center Interconnect Based on Multimode Fiber and Free-Space Optics. Optical Engineering, Vol. 60, 2021, No. 3, Art. No. 036113, doi: 10.1117/1.OE.60.3.036113.

[3] Iwagami, M.—Kumazawa, R.—Miyamoto, Y.—Ito, Y.—Ishimaru, M.—Morita, K.—Hamada, S.—Tamiya, N.—Yasunaga, H.: Risk of Cancer in Association with Ranitidine and Nizatidine vs Other H2 Blockers: Analysis of the Japan Medical Data Center Claims Database 2005–2018. Drug Safety, Vol. 44, 2021, No. 3, pp. 361–371, doi: 10.1007/s40264-020-01024-0.

[4] Liu, X.—Zhu, F.—Fu, Y.—Liu, Q.: A Resource Retrieval Method of Multimedia Recommendation System Based on Deep Learning. International Journal of Autonomous and Adaptive Communications Systems, Vol. 13, 2021, No. 4, pp. 400–418, doi: 10.1504/IJAACS.2020.112606.

[5] HajiAminShirazi, S.—Momtazi, S.: Cross-Lingual Embedding for Cross-Lingual Question Retrieval in Low-Resource Community Question Answering. Machine Translation, Vol. 34, 2020, No. 4, pp. 287–303, doi: 10.1007/s10590-020-09257-7.

[6] Li, H.: Internet Tourism Resource Retrieval Using PageRank Search Ranking Algorithm. Complexity, Vol. 2021, 2021, Art. No. 5114802, doi: 10.1155/2021/5114802.

[7] Zhao, G.—Ding, J.: Image Network Teaching Resource Retrieval Algorithm Based on Deep Hash Algorithm. Scientific Programming, Vol. 2021, 2021, Art. No. 9683908, doi: 10.1155/2021/9683908.

[8] Agersted, M. D.—Khodabandeloo, B.—Liu, Y.—Melle, W.—Klevjer, T. A.: Application of an Unsupervised Clustering Algorithm on in Situ Broadband Acoustic Data to Identify Different Mesopelagic Target Types. ICES Journal of Marine Science, Vol. 78, 2021, No. 8, pp. 2907–2921, doi: 10.1093/icesjms/fsab167.

[9] Louhichi, S.—Gzara, M.—Ben-Abdallah, H.: MDCUT2: A Multi-Density Clustering Algorithm with Automatic Detection of Density Variation in Data with Noise. Distributed and Parallel Databases, Vol. 37, 2019, No. 1, pp. 73–99, doi: 10.1007/s10619-018-7253-1.

[10] Gu, Z.—Li, P.—Lang, X.—Yu, Y.—Shen, X.—Cao, M.: A Multi-Granularity Density Peak Clustering Algorithm Based on Variational Mode Decomposition. Chinese Journal of Electronics, Vol. 30, 2021, No. 4, pp. 658–668, doi: 10.1049/cje.2021.03.001.

[11] Parmar, M. D.—Pang, W.—Hao, D.—Jiang, J.—Liupu, W.—Wang, L.—Zhou, Y.: FREDPC: A Feasible Residual Error-Based Density Peak Clustering Algorithm with the Fragment Merging Strategy. IEEE Access, Vol. 7, 2019, pp. 89789–89804, doi: 10.1109/ACCESS.2019.2926579.

[12] Siddique, M.—Samantara, T.—Mishra, S. P.: A Hybrid Prediction Model of Kernel Principal Component Analysis, Support Vector Regression and Teaching Learning Based Optimization Techniques. Current Journal of Applied Science and Technology, Vol. 40, 2021, No. 20, pp. 17–25, doi: 10.9734/cjast/2021/v40i2031460.

[13] Li, X.—Wu, H.—Yang, X.—Xue, P.—Tan, S.: Multiview Machine Vision Research of Fruits Boxes Handling Robot Based on the Improved 2D Kernel Principal Component Analysis Network. Journal of Robotics, Vol. 2021, 2021, Art. No. 3584422, doi: 10.1155/2021/3584422.

[14] Pan, C. S.—Mao, J. L.—Yang, Y.: Active Sonar Target-Echo Recognition Research Based on Deep Learning. Computer Simulation, Vol. 37, 2020, No. 11, pp. 179–183 (in Chinese).

[15] Xue, F.—Lu, W.—Chen, K.—Webster, C. J.: BIM Reconstruction from 3D Point Clouds: A Semantic Registration Approach Based on Multimodal Optimization and Architectural Design Knowledge. Advanced Engineering Informatics, Vol. 42, 2019, Art. No. 100965, doi: 10.1016/j.aei.2019.100965.

[16] Ahmed, N. S.—Ahmed, S. H.: Enhancement RC4 Algorithm Based on Logistic Maps with Multi-Parameters. Journal of Al-Qadisiyah for Computer Science and Mathematics, Vol. 11, 2019, No. 4, pp. 58–67.

Jiaming Jiang is a Senior Engineer. She received her Bachelor of economics from the Guangdong University of Foreign Studies in the Department of E-Commerce in 2010. Currently, she is working in the Guangdong Qingyuan Power Supply Bureau of the China Southern Power Grid. Her research interests include enterprise digital transformation and digital innovation application. She has published around 13 academic articles. Meanwhile, she has participated in 8 scientific research projects.

Guoheng Ruan is a Senior Engineer. He received his Bachelor degree in 2004 from the Sun Yat-Sen University in electronic information science and technology. He received his M.Sc. degree in 2009 from the South China University of Technology in control engineering. Now he is working in the Guangdong Qingyuan Power Supply Bureau of the China Southern Power Grid and his research areas include enterprise digital construction. He has published five academic articles. Meanwhile, he has presided over three scientific research projects.

Zhenggan Dai is working as a Senior Engineer. He received his Bachelor degree from the Hunan Institute of Science and Technology in the Department of Computer Science and Technology, 2008. Currently, he is working in the Guangdong Qingyuan Power Supply Bureau of the China Southern Power Grid. His research areas include enterprise digital transformation. He has published around three academic articles. Meanwhile, he has participated in four scientific research projects.

Computing and Informatics, Vol. 42, 2023, 943–964, doi: 10.31577/cai_2023_4_943

DYNAMIC MATCHING ALGORITHM OF HUMAN RESOURCE ALLOCATION BASED ON BIG DATA MINING

Yuping Yan

Guangdong Power Grid Co., Ltd.
Guangzhou 510000, China

Peiyao Xu*, Jianyong Wang

Guangdong Electric Power Information Technology Co., Ltd.
Guangzhou 510030, China
e-mail: xpy784574@163.com

Abstract. In order to ensure the dynamic matching effect of human resources allocation and improve the accuracy and efficiency of dynamic matching of human resources allocation, a dynamic matching algorithm of human resources allocation based on big data mining is studied. Analyze the meaning and function of big data mining, and explain the common analysis principles of big data mining. The information entropy is selected as the basis for measuring human resource allocation, the human resource allocation is extracted, and the similarity of human resource allocation is calculated using the Huasdorff similarity method based on time interpolation. According to the Apriori algorithm and FP-Growth classification algorithm, the human resource allocation is classified and mined, and the K-Means clustering algorithm is used to realize the dynamic matching of human resource allocation. The experimental results show that the proposed algorithm has better dynamic matching effect of human resources allocation, and can effectively improve the accuracy and efficiency of dynamic matching of human resources allocation.

Keywords: Big data mining, apriori algorithm, FP-growth classification algorithm, human resource allocation, K-means clustering algorithm, Huasdorff similarity method, dynamic matching

* Corresponding author

1 INTRODUCTION

With the integration of the world economy and China's entry into the WTO, the competition for talents will become more intense. Human resource is the most valuable and important resource among various resources of an enterprise, and it is the "first resource" for enterprise development. The combination and application of other resources in an enterprise must be promoted by human resources [1]. Across successive periods of the Industrial Revolution, organized labor, theory of management, cognitive economics, or social interactions, the idea of HRM has developed. As a result, the phrase Personnel Management has increasingly given way to the idea of HRM. However, the accumulation of human resources alone is not enough for an enterprise, and human resources must be allocated effectively and reasonably in order to maximize its benefits. The goals of human resource administration are to find qualified candidates, engage people, educate individuals, but then assist people in enhancing their productivity to enable them to accomplish great things and contribute to the company objectives. The allocation of enterprise human resources is to arrange all kinds of talents who meet the needs of enterprise development in the required positions in a timely and reasonable manner through assessment, selection, employment and training [2, 3]. To effectively utilise ability, increase the capabilities of employees, integrate it with other financial resources, and maximise the production of additional social and economic advantages for the business. Allocating human resources serves as both the beginning and the culmination of HR management. The types of HR allocation are workforce professional, Human Resource department trainee, HR professional, Human Resources supervisor. Its ultimate goal is to match individuals and positions, realize dynamic matching of human resource allocation, and improve the overall efficiency of the organization. The important element that decides whether the organization can develop consistently, steadily, and quickly is the amount of human resource allocation advantage, which has a direct impact on the prudent utilization of other assets or the entire allocating profit of the company. A continual flow of data and activity is HRM. The HRM element would suffer greatly from inaction. Keeping continuously mindful of how workers are performing, whether successfully there are doing everything, or how people think towards doing their duties is thus a crucial component of HRM. Therefore, the human resources management department must do a good job in the research of dynamic matching of human resources allocation according to the actual situation of the unit and the needs of the work tasks.

At present, scholars in related fields have carried out research on the dynamic matching of human resource allocation. [4] proposed a dynamic modeling algorithm for human resource allocation in construction projects. The distribution of human resources throughout execution of the project, as an efficient managed service or development element, can significantly affect program project in terms of quality and timeline. As a result, the assignment method can be made better by calculating labour requirements and comparing various human resource allocation plans.

Labor distribution is laborious due to the dynamic, complex relationships and feedback that exist within the project. A dynamic model for efficient labor allocation using a system dynamics approach is proposed. Using the model proposed in the current study enables the project to accurately estimate labor requirements and their efficient allocation, allowing for the necessary planning for the timely supply and distribution of project labor, both before and throughout project implementation. [5] proposed a business process human resource allocation algorithm based on team fault lines. This paper introduces the team fault line into the problem of human resource allocation. The resource characteristics are first analyzed from the demographic perspective and business process, and then the key characteristics are selected and the corresponding weights are determined using the information value. Secondly, qualitatively identify the team fault line according to the human resources clustering results, and quantitatively measure the strength and distance of the team fault line. Utilizing layered perspectives, basic or collective predictive algorithms are created. Following that, the assignment theory and procedure were created. The rationality and effectiveness of the method are evaluated through a real scenario, which can effectively allocate human resources and optimize business processes. Nevertheless, the aforementioned methods still struggle with a poor matched impact, low accuracy, or lack of efficiency.

Aiming at the above problems, a dynamic matching algorithm for human resource allocation based on big data mining is studied. The information entropy is used to extract the human resource configuration, and the Huasdorff similarity method based on time interpolation is used to calculate the similarity of human resource configuration. According to Apriori algorithm and FP-Growth classification algorithm, human resource allocation is classified and mined, and K-Means clustering algorithm is used to realize dynamic matching of human resource allocation. The dynamic matching effect of the algorithm is good, and it can dynamically match the accuracy and efficiency.

2 BIG DATA MINING TECHNOLOGY

2.1 Significance and Role of Big Data Mining

In today's world, no matter in the fields of economic operation, engineering construction, medical treatment, scientific research and invention or human resource management, a large amount of data is generated every day, and these data are very meaningful for obtaining valuable new discoveries. Through in-depth analysis of these data, people can better grasp people's needs and make accurate decisions. However, due to the large scale, complex content, and many data attributes involved in these data, traditional conventional methods have been unable to analyze these data in a timely and effective manner. Data analysis facilitates information discovery from unstructured, original data. Data mining methods enable information to be extracted from database systems as well as the centralized data or document store. However, big data mining technology can abstract some implicit and useful informa-

tion from it, and provide decision-makers with more accurate decision-making basis by discovering the correlation or law between different data.

The role of big data mining mainly has two aspects: one is to carry out task prediction, that is, to predict the value of a specific attribute according to some data attributes, so as to achieve the purpose of pre-judgment. The second is to describe the task, that is, to summarize the characteristics of some potential connections in the data through some unique patterns (such as correlation, trend, clustering, anomaly, etc.), so as to explore some characteristics, so as to obtain regularity [6, 7, 8].

2.2 Common Analysis Methods of Big Data Mining

According to the task requirements of big data mining, there are several commonly used analysis methods.

1. Predictive modeling: The model is mainly used to establish a model for the target variable by explaining the function of the variable. A quantitative logistic regression utilized for prediction evaluation is known as linear regression. Among the most basic and straightforward methods, its system analysis to demonstrate the connection among dependent variable. It mainly has two types of modeling tasks. One is classification, which is mainly used to predict discrete target variables. The second is regression, which is mainly used to predict continuous target variables.
2. Association analysis: It is mainly used to discover patterns of strongly associated features in data. The goal of association rule is to uncover intriguing connections within huge datasets. Both frequent item sets and clustering algorithms can be used to describe these fascinating interactions. A group of objects that commonly appear with others is called a frequent pattern combination. At its most basic stage, association rule extraction uses machine learning algorithms to search databases for similarities or co-occurrences in information. It detects common if-then relationships, that are the connection laws in and of itself. The patterns it finds are usually represented in the form of subsets with certain characteristics, and from large-scale data, it can extract the most interesting patterns in an efficient manner. Finding trends or correlations within a database that happen regularly is known as regular pattern mining in information gathering. This is often accomplished by searching through huge databases for things or groups of things that commonly occur simultaneously. It mainly discovers interesting associations or correlations between large transaction or relational datasets by mining frequent patterns, associations and correlations. By spotting patterns in the data, we may group objects that are highly connected collectively as well as quickly spot shared traits and correlations. Frequent pattern mining opens the door to independent research, such as grouping, categorization, or other information mining operations. The research focuses on mining frequent patterns and discovering effective association rules. Forecasting, correlation, or grouping are

the three broad subcategories into which data mining activities and structures can usually be divided. The main algorithms are Apriori algorithm, FP-growth algorithm, equivalence class transformation algorithm, etc. The Apriori algorithm uses database systems to establish frequent patterns and mine frequently occurring item sets. So soon as such item sets exist in the dataset frequently enough, it moves forward by detecting the frequently specific components or expanding those to progressively larger subsets. For frequent patterns mining, the FP-growth method is an enhanced variant of the Apriori method. It is a method for analysing sets of data to discover recurring connections or patterns.

3. Classification analysis: It is mainly used to determine which predefined target class the object belongs to. Its task is mainly to obtain an objective function f by learning, and then map each attribute set x to a predefined class label y. Classification generally has to go through two stages. The first stage is the learning stage, that is, the establishment of a classification model, which mainly uses the classification algorithm to construct a classifier through analysis and learning; the second stage is the classification stage, that is, using the previous learning. A classification model to predict the class label for the given data. The main classification methods are decision tree classification, rule-based classification, neural network, support vector machine and naive Bayes classification.
4. Cluster analysis: It is mainly used to divide data into meaningful or useful groups. Its task is to group data objects according to the information in the data that describes the objects and their relationships, and the objects within the group are similar (related) to each other. While objects in different groups are different (unrelated), the greater the similarity (correlation) within the group and the greater the difference between groups, the better the clustering. It has a wide range of applications in psychology, social sciences, biology, statistics, pattern recognition, information retrieval, machine learning and data mining. Clustering has different models, mainly hierarchical and divided, mutually exclusive, overlapping and fuzzy, complete and partial. The clusters clustered in practice can be divided into distinct separation, prototype-based, graph-based, density-based, and common property, etc., all of which are meaningful. There are three main types of clustering algorithms, namely K-Means algorithm (prototype-based, partitioned), agglomerative hierarchical clustering algorithm (graph-based), DBSCAN algorithm (density-based) and so on.
5. Anomaly detection: It mainly recognizes that the value of a special data point in the data is significantly different from the value of other data points, and has always achieved the purpose of early warning and prevention. An isolated data item that differs significantly from other values in the database is known as an outlier. It is a database abnormality that could have been brought about by a variety of mistakes in the collection, storage, or data manipulation. Because anomalies can distort general data patterns, anomaly detection systems are crucial in statistical. Outliers, also known as outliers, are points that are far away from other data points in the distribution map. Anomalous data objects

are unusual, or significantly inconsistent, so an analysis of outliers can reveal a bit of useful information. To look over thousands of transactions, classify, organize, or partition information in order to locate trends or identify theft, information extraction. In order to effectively identify unusual activity, neural nets acquire characteristics that appear suspect. It has a wide range of applications in fraud detection, intrusion detection, ecosystem imbalance detection, public health anomaly detection, and unusual case detection in healthcare systems.

2.3 Principle of Apriori Algorithm

Apriori is an algorithm for mining association rules in big data [9, 10]. Association rules are implication expressions in the form of $X \to Y$, where X and Y are disjoint item sets, namely $X \cap Y = \emptyset$. An associating policy's conviction or acceptance can be used to determine how strong it is. In contrast to Apriori, that searches the occurrences for every repetition, this method must search the dataset once. This approach is faster because it does not couple the elements. A compressed version of the information is kept in recollection. Support refers to the proportion of buying X and buying Y in the total number of item sets. The confidence level is to determine the proportion of Y buying both X and Y on the premise of buying X. The forms of support S and confidence C are defined as follows:

$$C(X \to Y) = \frac{\partial(X \cup Y)}{\partial(X)}, \tag{1}$$

$$C(X \to Y) = \frac{\partial(X \cup Y)}{\partial(X)}. \tag{2}$$

The principle of certainty, that is the proportion of the number of events that include the issue setting towards the quantity of transactions that have the antecedents, could be used to build frequent patterns after the common sets of items have been identified. The basic principle of Apriori is to generate an item set, denoted as L_1, by scanning the database for the first time and combining and decomposing. Then, through the L_1 item set, set the minimum support degree, continue to combine the items that satisfy the support degree, and obtain 2-item sets from 1-item set, denoted as L_2. Then, from 2-item sets, continue to generate 3-item sets recursively until no more frequent item sets are generated, and the whole process is over. The flow of the Apriori algorithm is as follows:

1. Set the minimum support threshold as $\min S$, and the credibility threshold as $\min C$;
2. Scan the database D, and set C_1 to represent the candidate item set, differentiate the frequent 1-item sets through $\min S$, and obtain the frequent 1-item sets L_1;
3. The candidate 2-item set C_2 is obtained after the L_1 operation, and then C_2 is divided by $\min S$ to obtain the frequent item set L_2;

4. Repeat the iteration until the largest frequent item set L_k is found to stop the iteration;
5. A strong rule equal to or exceeding $\min C$ is mined from the frequent item set L, and the rule is formulated as an association rule.

2.4 Principle of FP-Growth Algorithm

FP-Growth is a tree-based frequent item set mining algorithm. It recursively constructs a conditional FP-Tree and mines rule item sets on the conditional FP-tree [11, 12]. FP-Growth takes a divide and conquer approach, recursively transforming one problem into multiple problems. Store compressed transaction data along with important information about frequent item sets:

1. Contains a root node specified as empty, a set of item prefix subtrees and frequent item header tables as child nodes of the underlying node;
2. Each piece of data in the item header table records two parts, one part is the item set data, and the other part is the support count.

The running idea of FP-Growth algorithm is as follows:

1. For the frequent items to be mined, first construct a tree structure generated according to the transaction data list;
2. Construct the partition of the conditional pattern base for the generated tree structure. According to the item sets of the conditional pattern base, arrange the item sets that are greater than the minimum support count in the path to generate the conditional FP tree;
3. After the conditional FP tree is generated, the nodes under the previous path form a frequent item set, and the number of paths can be combined and connected with the previous path of the tree to obtain frequent items.

The FP-Growth algorithm obtains the number of items and support at the beginning of scanning the dataset. Sort the item sets in descending order of support. Then create the root node Root of the FP tree, and record it as empty. Then do the following operations for each item in the data set, and select the transaction item through the arrangement order of the item set. Then call the function inserted into the tree structure with the properly sorted transaction item table in the transaction and generate the result set.

The flow of the FP-Growth algorithm is as follows:

Step 1: Construct the FP tree:

1. Scan the database to count each item set;
2. Define the minimum support $\min S = 20\,\%$, that is, the minimum support (the minimum number of times the item appears) is 2;

3. Rearrange the item set in descending order. If there is an item less than 2, it needs to be deleted;
4. Readjust the item list in the database according to the number of item occurrences;
5. Build the FP tree: add the first record $(I2, I1, I5)$, add the second record $(I2, I4)$, and accumulate $(I2)$ when the same node appears. Add the third record $(I2, I3)$, the fourth record $(I2, I1, I4)$, the fifth record $(I1, I3)$, the sixth $(I2, I3)$, the seventh $(I1, I3)$, the eighth $(I2, I1, I3, I5)$, and the ninth $(I2, I1, I3)$, so the FP tree is established.

Step 2: Mining frequent item sets:

1. First, in the order from bottom to top, consider $I5$ first, get the conditional pattern bases $\langle(I2, I1 : 1)\rangle, \langle(I2, I1, I3 : 1)\rangle$, construct the FP tree, delete the nodes less than the support degree, form a single path and combine to get the $I5$ frequent item set $\{(I2, I5 : 2), (I1, I5 : 2), (I2, I1, I5 : 2)\}$;
2. Then consider $I4$, obtain the conditional pattern basis $\langle(I2, I1 : 1)\rangle, \langle(I2 : 1)\rangle$, construct the conditional FP tree, and obtain the $I4$ frequent item set $\{(I2, I4 : 2)\}$;
3. Then consider $I3$, and get the conditional pattern basis $\langle(I2, I1 : 2)\rangle, \langle(I2 : 2)\rangle, \langle(I1 : 2)\rangle$ to construct the conditional FP tree. Since this tree is not a single path, it is necessary to recursively mine $I3$ and consider $I3$ recursively. At this time, the conditional pattern base $\langle(I2 : 2)\rangle$ of $I1$ is obtained, that is, the conditional pattern base of $I1, I3$ is $\langle(I2 : 2)\rangle$, and the conditional FP tree is constructed to obtain the frequent item set $\{(I2, I3 : 4), (I1, I3 : 4), (I2, I1, I3 : 2)\}$ of $I3$;
4. Finally, consider $I1$, get the conditional pattern base $\langle(I2 : 4)\rangle$ to construct the conditional FP tree, and get the frequent item set $\{(I2, I1 : 4)\}$ of $I1$.

2.5 Principle of K-Means Clustering Algorithm

The K-Means algorithm is a well-known algorithm in the distance-based clustering algorithm. It uses distance as the evaluation index of similarity, and its principle is: if the vectors representing each point are close to each other in space, these points can be regarded as a class [13, 14]. That is, for a given sample set, when classifying, the sample set is divided into K classes according to the distance between samples, and the distance between points within a class is as small as possible, and the distance between classes is as large as possible. Max-min distance metric is used in K-Means grouping. Nevertheless, as the total of quadratic departures from the median is equivalent to the total of bilateral quadratic Euclidean values multiplied by the number of locations, K-Means is indirectly predicated on bilateral Euclidean similarities among sample points.

The K-Means algorithm has a basic assumption for the data that needs to be clustered: for each class, a center point can be selected so that the distance from

all points in the class to the center point is less than the distance to the centers of other classes. However, in the actual classification, the obtained data often cannot directly meet such requirements, and can only be as close as possible. The reasons for such differences are often inherent in the data itself, or the data can no longer be classified.

The method of the K-Means algorithm is to select K initial reference points according to the input parameter K, and divide all the points in the data set into K classes according to the K reference points. Given that movement is subjective by design, frames of reference are crucial since they accurately express an individual's attitude. It makes sense that the more optimally placed such initially cluster centers are placed, the less repetitions of the K-Means classification methods will be required to reach an ideal set of anchor nodes or grouping participation according to the distance from such anchor nodes. The centroid of these K classes (the average of all points in the class) is used as the reference point for the next iteration, and the dataset is divided into final K classes through continuous iterative updates. The iteration makes the chosen reference points get closer and closer to the true class centroid, consequently, the clustering effect is improving.

Let the Equation (3) q dimensional data set

$$W = \{w_i \mid w_i \in R^q, i = 1, 2, \ldots, N\} \tag{3}$$

be aggregated into K classes α_1, α_2, ..., α_K, and their centroids are in turn z_1, z_2, ..., z_K, where:

$$z_i = \left(\frac{1}{m_i}\right) \sum_{w \in \alpha_K} w. \tag{4}$$

In Equation (4), m_i is the number of data points in class α_i. The quality of the clustering effect is represented by the objective function E:

$$E = \sum_{i=1}^{k} \sum_{j=1}^{m_i} b_{ij}\left(w_j, z_i\right). \tag{5}$$

In Equation (5), $b_{ij}\left(w_j, z_i\right)$ is the Euclidean distance between w_j and z_i. The objective function E is actually the sum of the distances between each data point and the centroid of the class, so the smaller the E value, the more compact the class is. Therefore, the algorithm can seek a good clustering method by continuously optimizing the value of E. When E takes a minimum value, the corresponding clustering method is the optimal method.

The steps of the K-Means algorithm are as follows:

Step 1: randomly select K initial reference points z_1, z_2, ..., z_K from W;

Step 2: Use z_1, z_2, ..., z_K as a reference point to divide W. The division is based on the following principles: If $b_{ij}\left(w_i, z_j\right) < b_{in}\left(w_i, z_n\right)$, among them, $n = 1, 2, \ldots, K$; $j = 1, 2, \ldots, K$; $j \neq n$; $i = 1, 2, \ldots, N$, then divide w_i into class α_j;

Step 3: According to Equation (4), recalculate the centroid z_1^*, z_2^*, ..., z_K^* of the class;

Step 4: If $z_i^* = z_i$ is true for any $i \in \{1, 2, \ldots, K\}$, the algorithm ends, and the current z_1^*, z_2^*, ..., z_K^* represents the final class; Otherwise, let $z_i^* = z_i$ go back to step 2 for execution.

In order to prevent the infinite loop from occurring because the termination condition in step 4 cannot be satisfied, a maximum number of iterations is usually set in the algorithm, or a fixed threshold β is set. The algorithm is considered to end when there is $|z_i^* - z_i| < \beta$ for all z_i. The iterative process of the K-Means algorithm is shown in Figure 1.

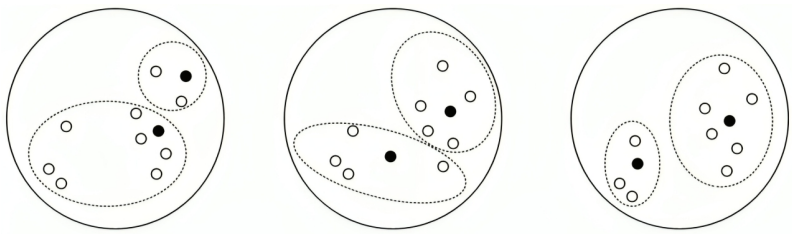


Figure 1. Iterative process of K-Means algorithm

The time complexity of the K-Means algorithm is $O(\chi KN)$. Where χ is number of algorithm iterations, K is the number of categories, and N is the number of data points in the dataset. The size of k, that must always be provided in order to conduct any grouping investigation, determines how the K-Means method operates. Ultimately, grouping with various input parameters will yield various outcomes. Many factors can affect the performance of the K-Means algorithm, including the number of cluster categories K, the choice of initial reference points, and the type of input data. Therefore, it is necessary to correctly determine the initial reference point according to the specific application, and select a similarity measurement strategy that is consistent with the data type, so as to ensure that the K-Means algorithm obtains better performance.

3 DYNAMIC MATCHING ALGORITHM OF HUMAN RESOURCE ALLOCATION

3.1 Extracting Human Resource Configuration

In order to effectively realize the dynamic matching of human resource allocation and further reduce the amount of data in the matching process, information entropy is selected as the basis for measuring human resource allocation, and an appropriate amount of human resource allocation is obtained for subsequent analysis and processing of the similarity of the allocation. Data minimization can save collection

expenses while also increasing storing speed and economy. Feature extraction employs a variety of techniques to lessen the amount of information that is kept on the device.

Using information entropy as an evaluation parameter, the human resource allocation at this time has more characteristic information for human resource allocation analysisUtilizing HR information enables us to develop a company or its individual divisions in the ways that we desire it to expand. Our firm will expand responsibly and employ a content staff with the support of thorough HR dataset obtained or a solid method of marketing. In particular, information entropy can also be used to measure the amount of information about human resource allocation. It is believed here that when the amount of information on human resource allocation is more, that is, the greater the number of different human resource allocations, it has more reference value in the analysis of human resource allocation. Whenever all possible benefits are taken into account, data entropy is typically defined as the average quantity of data that an occurrence conveys.

Human resource allocation is divided by time and there is

$$R = \{SubR_1, SubR_2, \ldots, SubR_5\}.$$

Take one of the sub-human resources allocation $SubR_a$ $(1 \leq a \leq S)$ as an example in Equation (6):

$$\mathrm{SubR_a} = \{y_{k+1}, y_{k+2}, \ldots, y_{k+r}\}. \tag{6}$$

There are r individual human resource configuration points in this sub-human resource configuration. Among them, different human resource allocation values are $U = \{u_1, u_2, \ldots, u_e\}$, the number of occurrences of each human resource allocation value is $USum$. The calculation formula is as follows:

$$P_i = \frac{U\,\mathrm{Sum}_i}{\mathrm{sum}_j}\ (1 \leq i \leq \theta). \tag{7}$$

In Equation (7), $USum$ is the quantity of the human resource configuration in the sub-human resource configuration, sum_j is the total number of human resource configuration points in the sub-human resource configuration, and e is the number of non-repetitive human resource configuration in the subhuman resource configuration. The degree of surprise (or uncertainty) associated with the quantity of a stochastic process or the result of a randomized operation is measured by entropy values, often known as Shannon's entropy. Its importance in the tree structure comes from the fact that it enables us to calculate the diversity or impurities of the target attribute. Joint Entropy and Conditional Probability are the two kinds. The information entropy of the human resource allocation can be obtained by the frequency of the above human resource allocation points, as shown in Equation (8):

$$H_j = -\sum_{i=1}^{e} P_i \log_2 P_i. \tag{8}$$

When each human resource configuration has the corresponding information entropy, extract the information entropy $H > 0$, and a total of H_s human resource configurations. And according to the size of the information entropy, the extraction percentage δ of human resource allocation is given, and the total number of human resource allocation H_n to be extracted can be calculated and retained. The relevant formula is as follows in Equation (9):

$$H_n = H_s \times \delta. \tag{9}$$

The human resource configuration with the quantity H_n extracted from the above is sorted according to the time period number and retained.

3.2 Calculate the Similarity of Human Resource Allocation

On the basis of the above-mentioned extraction of human resource configuration, the dynamic matching accuracy of human resource configuration is high. Therefore, using the Huasdgrff similarity method based on time interpolation, the similarity of human resource allocation is calculated. Training personnel and coordinating their goals for personal growth with the overarching objectives of the organisation or company are the main concerns of developing human resources. The administration of human resources places more of an emphasis on conformity, insurance, pay, and labour rights.

In order to improve the dynamic matching accuracy of human resource allocation, a time interpolation method is introduced to supplement human resource allocation. Non-human assets are material goods or items that are present without the presence of individuals. Humans are able to view, interact with, and use them. Material wealth are another name for non-human elements. Vehicles, clinics, schools, libraries, playgrounds, gasoline, laptops, magazines, calendars, flowers, and cash are a few instances. From the target human resource configuration R_A and the matching human resource configuration R_B, under each human resource configuration time period, two non-empty human resource configurations $SubR_A$ and $SubR_B$ are extracted. Retain the time when the $SubR_A$ non-empty sub-HR configuration appears. For segments in the chronology whose frequency deviates from the sequencing parameters, time approximation is used. When transferring the schedule in its entirety to a frequency other than the series parameters, temporal compression is used. The possible positions of $SubR_B$ non-kongzi human resource allocation under these times are calculated by the $SubR_B$ non-kongzi human resource allocation function, that is, the interpolation point. Similarly, keep the time when the $SubR_B$ non-empty sub-human resource configuration appears, and calculate the position where the $SubR_A$ non-empty sub-human resource configuration appears under these times. The extracted human resource configuration needs to complete the above interpolation operation.

If an opponent selects a location within one of the different pairs, from which one should subsequently move to the second setting, the Hausdorff duration is the

greatest length that can be required of someone. It represents the largest possible distance between a location in one set and its nearest neighbour in the opposite set. Using the Hausdorff distance method [15], the human resource allocation after time interpolation is matched with the target human resource allocation by similarity. One can gain the following skills through working as a HRM expert: understanding of regional, state wide, or national employment regulations; knowledge of the Office Software Suite and personnel software solutions, understanding of experienced experts and planning tools in the field. According to the concept, the operational plan and HR systems must be handled in a manner that is in line with the corporate objectives. Add up the Haysdorff distances obtained between each corresponding human resource configuration, and obtain the mean value as the final distance between human resource configurations. The specific formula is as follows:

$$H_k\left(SubR_A, SubR_B\right) = \max\left(h_k\left(SubR_A, SubR_B\right)\right), \left(h_k\left(SubR_B, SubR_A\right)\right). \quad (10)$$

In Equation (10), $H_k\left(SubR_A, SubR_B\right)$ is the Hausdorff distance between $SubR_A$ and $SubR_B$ of nonempty human resource allocation, $h_k\left(SubR_A, SubR_B\right)$ is the one-way Hausdorff distance from $SubR_A$ to $SubR_B, h_k\left(SubR_B, SubR_A\right)$ is the one-way Hausdorff distance from $SubR_B$ to $SubR_A$. Through the above process, the similarity of human resource configuration is calculated accordingly. The closest locations will possess the smallest distances whenever evaluating through length; however, the closest locations will display the greatest similarities when comparing by resemblance.

3.3 Classification and Mining of Human Resource Allocation

The implemented to achieve of human resource distribution is separated into four intervals following the comparability of human resource allocation calculation in order to enable categorization and extraction of personal allocation of resources

$$u_1 : 0, 5), u_2 : 5, 10), u_3 : 10, 15), u_4 : 10, +\infty).$$

Finding the economic indicators among the provided datasets is made easier by doing this. To calculate the derivative of a variable for an intermediary variable of the independence variable, this procedure is usually required. A daemon in the local scheduler records time slice information for human resource configuration. The human resource configuration transaction set is shown in Table 1.

In Table 1, (d_1, d_2, d_3) represents the time slice information of human resource configuration, u_{ij} represents the time slice interval of human resource configuration i is j, and the time slice without computing tasks is marked as u_{i1}, where i is the human resource configuration number.

According to the Apriori algorithm and the FP-Growth classification algorithm, the transactions in Table 1 are mined, and the minimum support count $\min S = 2$

ID	d_1	d_2	d_3	User ID
1	u_{11}	u_{22}	u_{31}	c_1
2	u_{12}	u_{21}	u_{32}	c_2
3	u_{14}	u_{21}	u_{31}	c_1
4	u_{11}	u_{21}	u_{31}	c_1
5	u_{12}	u_{22}	u_{33}	c_2
6	u_{13}	u_{22}	u_{34}	c_3
7	u_{12}	u_{22}	u_{32}	c_2

Table 1. Human resource configuration transaction set

is set, and the frequent pattern is

$$\{\langle u_{11}, u_{31}, c_1\rangle, \langle u_{12}, u_{22}, c_2\rangle, \langle u_{12}, u_{32}, c_2\rangle, \langle u_{21}, u_{31}, c_1\rangle\}.$$

Set the minimum confidence min $C = 70\,\%$ to generate the classification rules as Equations (11) and (12):

$$u_{11} \wedge u_{31} \Rightarrow c_1, u_{11} \wedge c_1 \Rightarrow u_{31}, u_{11} \Rightarrow u_{31} \wedge c_1, u_{21} \wedge u_{31} \Rightarrow c_1, u_{21} \wedge c_1 \Rightarrow u_{31}, \quad (11)$$

$$u_{12} \wedge u_{22} \Rightarrow c_2, u_{22} \wedge c_2 \Rightarrow u_{12}, u_{12} \wedge u_{32} \Rightarrow c_2, u_{32} \wedge c_2 \Rightarrow u_{12}, u_{32} \Rightarrow c_2 \wedge u_{12}. \quad (12)$$

According to the correspondence between u_{ij} and d_i, it is concluded that the time slice of c_1 using d_1 and d_2 is u_1 or the time slice using d_2 and d_3 is $u_1 \cdot c_2$ uses d_1 and d_3 time slices for u_2 or d_1 and d_2 time slices for u_2.

A data gathering systems could also be categorized according to the types of information mining, expertise mineable, methodologies used, and systems adopted. At the same time, classification and mining can be carried out according to the configuration of human resources, and classification rules based on configuration can be generated, and the classification rules can be obtained as follows:

$$u_2 \wedge c_1 \Rightarrow u_1, u_1 \wedge c_2 \Rightarrow u_2, u_2 \wedge c_2 \Rightarrow u_1, c_2 \Rightarrow u_1 \wedge u_2. \quad (13)$$

That is, if c_1 uses the u_2 task request, c_1 will also use the u_1 task. c_2 uses both types of time slices u_1 and u_2 when requesting human resource allocation.

3.4 Realize Dynamic Matching of Human Resource Allocation

After classifying and mining human resource allocation, K-Means clustering algorithm is used to realize dynamic matching of human resource allocation. Suppose there are g configurations in human resources, which are p_1, p_2, ..., p_g, respectively, each configuration has h_i task requests, and the time length of task requests is denoted as U_{ij}. Among them, i is the human resources configuration number, j is the task number, and $1 \leq i \leq g, 1 \leq j \leq h_i$. From g, h_i and U_{ij}, it can be concluded that the total length of time slice U_s for all human resource allocation and all task

requests is expressed by Equation (14):

$$U_s = U_{11} + U_{12} + \cdots + U_{1h_1} + \cdots + U_{g1} + U_{g2} + \cdots + U_{gh_g}. \tag{14}$$

Suppose there are k departments in the enterprise, which are responsible for providing corresponding human resource allocation, which are $l_1, l_2, \ldots, l_k$ respectively. Each department contains V_i personnel, and each personnel includes ε isomorphic human resource allocations. The total number of human resource allocations B_s in the entire enterprise is in Equation (15):

$$B_s = \varepsilon \times (V_1 + V_2 + \cdots + V_k). \tag{15}$$

Through classification mining, the human resource allocation of the same department can be classified together, resulting in the following rules:

$$c_i \wedge \cdots \wedge c_j \Rightarrow l_\phi. \tag{16}$$

In Equation (16), $1 \leq i \leq k$, $1 \leq j \leq g$, $1 \leq \phi \leq k$. Based on the classification rules and mining patterns obtained by the classification mining algorithm, the K-Means clustering algorithm is used to submit the grouped tasks to the corresponding personnel, and distribute them to the personnel corresponding to the human resource configuration according to the physical location of the task. In the human resource allocation B_s in the whole enterprise, $f_r = \sum_{i=l_\phi} f_i$, that is, after any one task is completed, it means that the human resource allocation is insufficient. When $f \sum_{i=l_\phi} f_{i_{\max}}$, the corresponding personnel will execute within B_s with the closest adjacent distance, and the remaining tasks will be handed over to B_s for dynamic matching. Through the above steps, dynamic matching of human resource allocation is realized.

4 EXPERIMENTAL ANALYSIS

4.1 Experimental Environment and Data

In order to verify the effectiveness of the dynamic matching algorithm of human resource allocation based on big data mining, the experimental platform adopts Thinksad L430 loaded with Windows 7 system, the memory is 4 GB, and the CPU is Intel Core i7 with 2.9 GHz. Non-relational documents library MongoDB supports retention which is similar to JSON. The MongoDB system has complete querying capabilities, persistence, as well as a configurable schema that makes it possible to hold complex data. It also features comprehensive and user-friendly APIs. The experiment uses Java language to operate MongoDB for data processing and analysis, and uses Java and Matlab to implement the proposed algorithm. Multiple information scientific operations, such as data gathering, comprising information importing, metadata management, deep learning, scientific techniques, Natural Language Pro-

cessor (NLP), or data visualisation, make extensive use of Java. The main parameter settings are shown in Table 2.

Parameter	Numerical Value
Number of departments in the enterprise	$k = 3$
Number of people in each department	$V_i = 5$
Number of processors per node The number of homogeneous human resource allocations in each person	$\varepsilon = 8$
The total number of human resource allocations in the entire enterprise	$B_s = 120$

Table 2. Main parameter settings

The algorithm of reference [4], the algorithm of reference [5] and the proposed algorithm are respectively used to compare the dynamic matching resource utilization, dynamic matching accuracy and dynamic matching time of different algorithms, so as to verify the effectiveness of the proposed algorithm.

4.2 Comparison Results of Dynamic Matching Effect of Human Resource Allocation

In order to verify the dynamic matching effect of human resource allocation of the proposed algorithm, the dynamic matching resource utilization rate is taken as the evaluation index. The higher the dynamic matching resource utilization rate is, the better the dynamic matching effect of the algorithm is. The algorithm of reference [4], the algorithm of reference [5] and the proposed algorithm are used to compare, and the comparison results of the resource utilization ratio of dynamic matching of human resource allocation of different algorithms are obtained as shown in Figure 2.

According to Figure 2, when the total number of human resource allocations in the entire enterprise reaches 120, the average human resource allocation dynamic matching resource utilization rate of the algorithm of reference [4] is 89.3 %, the average human resource allocation dynamic matching resource utilization rate of the algorithm of reference [5] is 81.2 %. The average human resource allocation dynamic matching resource utilization rate of the proposed algorithm is as high as 98.7 %. It can be seen that the dynamic matching of human resource allocation of the proposed algorithm has a higher utilization rate of resources, indicating that the dynamic matching of human resources allocation of the proposed algorithm has a better effect.

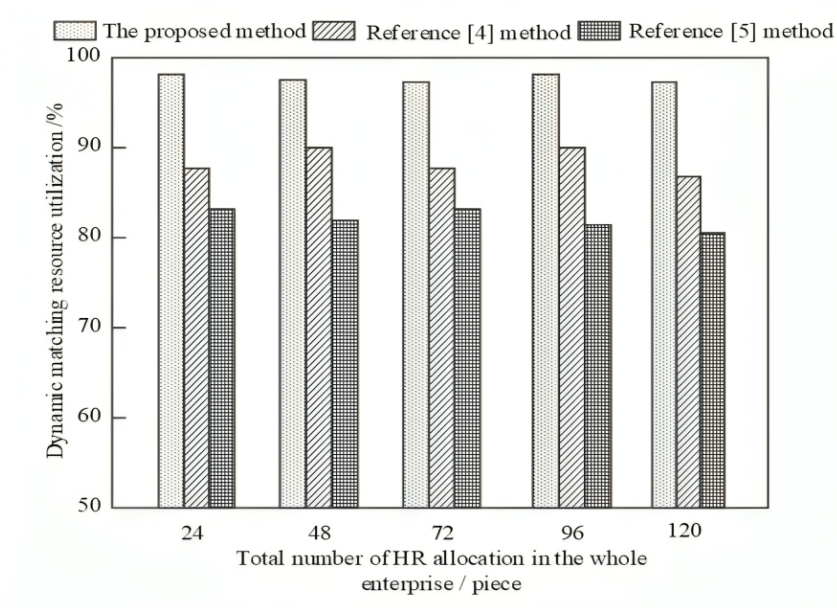


Figure 2. Comparison results of resource utilization ratio of dynamic matching of human resource allocation with different algorithms

4.3 Comparison Results of Dynamic Matching Accuracy of Human Resource Allocation

On this basis, the dynamic matching accuracy of human resource allocation of the proposed algorithm is further verified, and the dynamic matching accuracy is used as the evaluation index. The higher the dynamic matching accuracy, the higher the dynamic matching accuracy of the algorithm's human resource allocation. The algorithm of reference [4], the algorithm of reference [5] and the proposed algorithm are used to compare, and the comparison results of the dynamic matching accuracy of human resource allocation of different algorithms are shown in Figure 3.

According to Figure 3, when the total number of human resource allocations in the entire enterprise reaches 120, the average dynamic matching accuracy of human resource allocation of the algorithm of reference [4] is 88.5 %, the average human resource allocation dynamic matching accuracy of the algorithm of reference [5] is 80.1 %. The average dynamic matching accuracy rate of human resource allocation of the proposed algorithm is as high as 95.8 %. It can be seen that the dynamic matching accuracy of human resource allocation of the proposed algorithm is high, indicating that the dynamic matching accuracy of human resource allocation of the proposed algorithm is high.

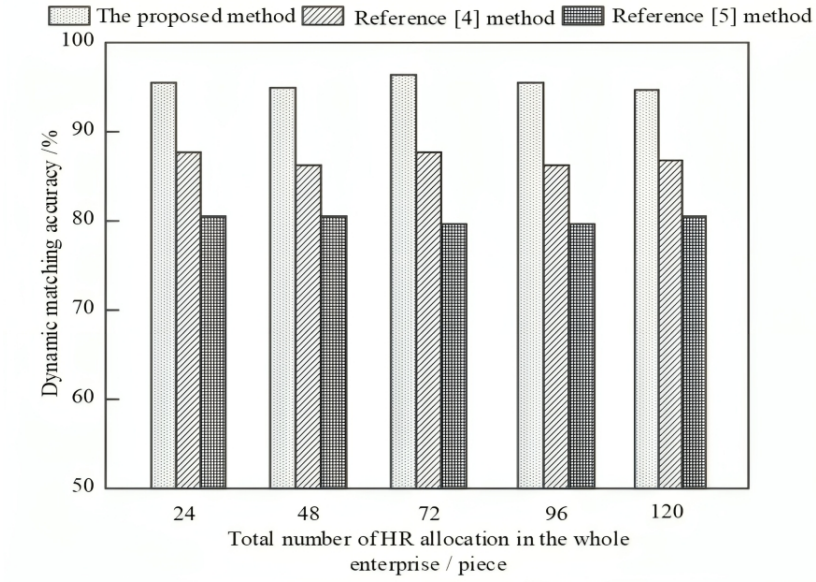


Figure 3. Comparison results of dynamic matching accuracy of human resource allocation of different algorithms

4.4 Comparison Results of Dynamic Matching Efficiency of Human Resource Allocation

The dynamic matching efficiency of human resource allocation of the proposed algorithm is further verified, and the dynamic matching time is used as the evaluation index. The process is check out the present personnel, establish a transition strategy, make a decision on how to expand assets going forward, make plans for employee training, besides conduct a gap assessment. The shorter the dynamic matching time is, the higher the dynamic matching efficiency of human resource allocation of the algorithm is. The algorithm of reference [4], the algorithm of reference [5] and the proposed algorithm are used to compare, and the comparison results of the dynamic matching time of human resource allocation of different algorithms are shown in Table 3. According to the concept, the administrative architecture or HR department must be handled in a manner that is in line with the corporate objectives. The approach is employed to make it easier to fulfil the organization's objectives in the areas of production, profitability, or effectiveness.

According to the data in Table 3, with the increase of the total number of human resource allocation in the whole enterprise, the dynamic matching time of human resource allocation of different algorithms gradually increases. When the total number of human resource allocations in the entire enterprise reaches 120,

Total Number of HR Allocation in the Whole Enterprise [Piece]	The Proposed Algorithm [s]	The Algorithm of Reference [4] [s]	The Algorithm of Reference [5] [s]
24	0.8	2.8	4.9
48	1.9	4.6	6.2
72	2.6	6.9	8.9
96	3.5	8.8	10.2
120	5.1	10.2	12.6

Table 3. Comparison results of dynamic matching time of human resource allocation with different algorithms

the dynamic matching time of human resource allocation of the algorithm of reference [4] is 10.2 s, and the dynamic matching time of the algorithm of reference [5] is 12.6 s. The dynamic matching time of human resource allocation of the proposed algorithm is only 5.1 s. It can be seen that the dynamic matching time of human resource allocation of the proposed algorithm is short, indicating that the dynamic matching efficiency of human resource allocation of the proposed algorithm is high.

5 CONCLUSION

This paper studies the dynamic matching algorithm of human resource allocation based on big data mining, and adopts the relevant algorithms of big data mining to realize the dynamic matching of human resource allocation. The algorithm has better dynamic matching effect of human resource allocation, and can effectively improve the accuracy and efficiency of dynamic matching of human resource allocation. However, the algorithm is only used for simulation experiments of a small amount of human resource allocation, and does not consider the situation of massive human resource allocation. Therefore, in the following research, further research on the allocation of massive human resources is needed.

6 DECLARATIONS

Funding: No funds, grants were received by any of the authors.

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

Data availability: All data generated or analysed during this study are included in the manuscript.

Code availability: Not applicable.

Authors' contributions: All authors contributed to the design and methodology of this study, the assessment of the outcomes and the writing of the manuscript.

REFERENCES

[1] Monyei, E. F.—Agbaeze, K. E.—Isichei, E. E.: Organisational Paranoia and Employees' Commitment: Mediating Effect of Human Resources Policies. International Journal of Scientific and Technology Research, Vol. 9, 2020, pp. 5172–5185.

[2] Xu, J.—Wang, B.—Min, G.: Research on Human Resource Allocation Model Based on SOM Neural Network. Research Anthology on Human Resource Practices for the Modern Workforce, IGI Global, 2022, pp. 513–525, doi: 10.4018/978-1-6684-3873-2.ch027.

[3] Kieling, E. J.—Rodrigues, F. C.—Filippetto, A.—Barbosa, J.: Smartalloc: A Model Based on Machine Learning for Human Resource Allocation in Projects. Proceedings of the 25th Brazillian Symposium on Multimedia and the Web (WebMedia'19), 2019, pp. 365–368, doi: 10.1145/3323503.3360643.

[4] Dabirian, S.—Abbaspour, S.—Khanzadi, M.—Ahmadi, M.: Dynamic Modelling of Human Resource Allocation in Construction Projects. International Journal of Construction Management, Vol. 22, 2022, No. 2, pp. 182–191, doi: 10.1080/15623599.2019.1616411.

[5] Zhao, W.—Pu, S.—Jiang, D.: A Human Resource Allocation Method for Business Processes Using Team Faultlines. Applied Intelligence, Vol. 50, 2020, No. 9, pp. 2887–2900, doi: 10.1007/s10489-020-01686-4.

[6] Chamikara, M. A. P.—Bertok, P.—Liu, D.—Camtepe, S.—Khalil, I.: Efficient Privacy Preservation of Big Data for Accurate Data Mining. Information Sciences, Vol. 527, 2020, pp. 420–443, doi: 10.1016/j.ins.2019.05.053.

[7] Fernandez-Basso, C.—Ruiz, M. D.—Martin-Bautista, M. J.: A Fuzzy Mining Approach for Energy Efficiency in a Big Data Framework. IEEE Transactions on Fuzzy Systems, Vol. 28, 2020, No. 11, pp. 2747–2758, doi: 10.1109/TFUZZ.2020.2992180.

[8] Fernandez-Basso, C.—Francisco-Agra, A. J.—Martin-Bautista, M. J.—Ruiz, M. D.: Finding Tendencies in Streaming Data Using Big Data Frequent Itemset Mining. Knowledge-Based Systems, Vol. 163, 2019, pp. 666–674, doi: 10.1016/j.knosys.2018.09.026.

[9] Zhang, R.—Chen, W.—Hsu, T. C.—Yang, H.—Chung, Y. C.: ANG: A Combination of Apriori and Graph Computing Techniques for Frequent Itemsets Mining. The Journal of Supercomputing, Vol. 75, 2019, No. 2, pp. 646–661, doi: 10.1007/s11227-017-2049-z.

[10] Dharshinni, N. P.: Analysis of Definite Integral Material Topics for Improve Student Learning Using Apriori Algorithm. Journal of Informatics and Telecommunication Engineering, Vol. 4, 2021, No. 2, pp. 294–300, doi: 10.31289/jite.v4i2.4316 (in Indonesian).

[11] Mahrousa, Z.—Alchawafa, D. M.—Kazzaz, H.: Frequent Itemset Mining Based on Development of FP-Growth Algorithm and Use MapReduce Technique. Association of Arab Universities Journal of Engineering Sciences, Vol. 28, 2021, No. 1, pp. 83–98, doi: 10.33261/jaaru.2021.28.1.008.

[12] Firmansyah, F.—Yulianto, A.: Market Basket Analysis for Books Sales Promotion Using FP Growth Algorithm, Case Study: Gramedia Matraman Jakarta. Journal of Informatics and Telecommunication Engineering, Vol. 4, 2021, No. 2, pp. 383–392, doi: 10.31289/jite.v4i2.4539 (in Indonesian).

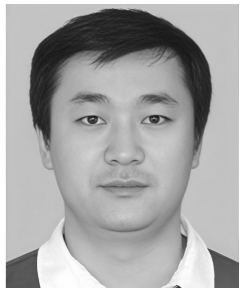
[13] Huyan, J.—Li, W.—Tighe, S.—Deng, R.—Yan, S.: Illumination Compensation Model with K-Means Algorithm for Detection of Pavement Surface Cracks with Shadow. Journal of Computing in Civil Engineering, Vol. 34, 2020, No. 1, Art. No. 04019049, doi: 10.1061/(ASCE)CP.1943-5487.0000869.

[14] Geng, D. Z.—Xu, Q.: High-Dimensional Mixed Attribute Data Mining Method Based on K-Means Clustering Algorithm. Computer Simulation, Vol. 38, 2021, No. 2, pp. 308–312 (in Chinese).

[15] Kraft, D.: Computing the Hausdorff Distance of Two Sets from Their Distance Functions. International Journal of Computational Geometry and Applications, Vol. 30, 2020, No. 1, pp. 19–49, doi: 10.1142/S0218195920500028.

Yuping Yan is a Senior Engineer. In 2007, he graduated from the Sun Yat Sen University, majoring in computer science. In 2010, he graduated from the Sun Yat Sen University with a master's degree in computer software engineering. His research field involves data mining. He has published 24 academic articles, participated in 145 scientific research projects and obtained 16 invention patents in the field of digital grid. He has won 45 awards such as the Company's Science and Technology Progress Award and Management Innovation Award. He is now working in the Guangdong Power Grid Co., Ltd., responsible for the company's digital transformation construction.

Peiyao Xu is an Intermediate Engineer. She graduated from the Guangdong University of Technology with a bachelor's degree in 2010, majoring in computer science and technology. Her research fields involve information system operation and maintenance and digital construction. She has published 2 academic articles, has participated in 2 scientific research projects, won 1 invention patent in the field of digital grid, and won 1 science and technology progress award of the company. She is now working in the China Southern Power Grid Digital Enterprise Technology (Guangdong) Co., Ltd., responsible for the company's digital construction.



Jianyong Wang is a Senior Engineer. He graduated from the China University of Mining and Technology in 2004, majoring in computer science and technology. In 2015, he graduated from the Wuhan University with a master's degree in software engineering. His research fields involve power informatization and software development. He has published 14 academic articles, has participated in 50 scientific research projects and obtained 10 invention patents in the field of digital power grid. He has won 10 awards such as the Company's Science and Technology Progress Award, Technology Improvement Award and Management Innovation Award. He is currently working in the China Southern Power Grid Digital Enterprise Technology (Guangdong) Co., Ltd., responsible for the company's digital transformation construction.

Computing and Informatics, Vol. 42, 2023, 965–992, doi: 10.31577/cai_2023_4_965

ENSEMBLE BASED FEATURE EXTRACTION AND DEEP LEARNING CLASSIFICATION MODEL WITH DEPTH VISION

Kumari Priyanka Sinha*

Department of Computer Science and Engineering
Nalanda College of Engineering, Chandi
Bihar, India
e-mail: kumaripriyankas@outlook.com

Prabhat Kumar, Rajib Ghosh

Department of Computer Science and Engineering
National Institute of Technology Patna
Patna, India
e-mail: {Prabhat, rajib.ghosh}@nitp.ac.in

Abstract. It remains a challenging task to identify human activities from a video sequence or still image due to factors such as backdrop clutter, fractional occlusion, and changes in scale, point of view, appearance, and lighting. Different appliances, as well as video surveillance systems, human-computer interfaces, and robots used to study human behavior, require different activity classification systems. A four-stage framework for recognizing human activities is proposed in the paper. As part of the initial stages of pre-processing, video-to-frame conversion and adaptive histogram equalization (AHE) are performed. Additionally, watershed segmentation is performed and, from the segmented images, local texton XOR patterns (LTXOR), motion boundary scale-invariant feature transforms (MoBSIFT) and bag of visual words (BoW) based features are extracted. The Bidirectional gated recurrent unit (Bi-GRU) and the Bidirectional long short-term memory (Bi-LSTM) classifiers are used to detect human activity. In addition, the combined decisions of the Bi-GRU and Bi-LSTM classifiers are further fused, and their accuracy levels are determined. With this Dempster-Shafer theory (DST)

* Corresponding author

technique, it is more likely that the results obtained from the analysis are accurate. Various metrics are used to assess the effectiveness of the deployed approach.

Keywords: Human activities, improved LTXOR, BoW, Bi-LSTM, Bi-GRU classifier

Mathematics Subject Classification 2010: 46-T30

1 INTRODUCTION

Human action recognition (HAR) is a classification task in which the movement of a person is evaluated using data from different sources, such as sensors and cameras. It has several applications in the health care industry, primarily, in monitoring the actions of elderly people and detecting falls [1, 2, 3]. The system has the potential to support innovative appliances, such as enhanced realism, internet of things (IoT), interior localization, and smart building control systems to maintain a secure indoor environment with superior energy efficiency [4, 5]. Several studies have been conducted using computer vision (CV), smartphones, sensors, and ambient devices to develop HAR schemes. There are two types of monitoring schemes: passive and active [6, 7, 8]. Wearable cameras and sensors are used as part of active monitoring systems (AMS). A sensor-based HAR requires individuals to hold sensors, such as gyroscopes, accelerometers, and pedometers, which can be difficult to manage in several situations, particularly for senior citizens [9, 10, 11].

With a vision-oriented HAR [12, 13], humans can be recognized both indoors and outdoors using cameras and machine vision schemes. One of the major challenges that such schemes face is the presence of noise in the video and image streams [14, 15, 16]. Depending on the surrounding environment, the capturing device, and several other factors, there may be several types of noise, including multiplicative noise, additive noise, etc. If a vision-based recognition and classification scheme is to be effective, it must eliminate the noise before implementation. In recent studies, deep learning schemes, such as deep belief networks (DBNs), convolutional neural networks (CNNs), and deep convolutional neural networks (DCNNs) have produced competent results in image-oriented HAR [17, 18, 19]. Although these improvements have been made, certain challenges remain, including a lack of accuracy, that must be overcome by the application of appropriate technology in the future [20, 21, 22, 23]. The major contributions of this study are listed as follows.

1. The first step in this study is to propose a multi-feature fusion approach that will combine local texton XOR pattern (LTXOR), BoW, and motion boundary scale-invariant feature transform (MoBSIFT) features to investigate datasets at a deeper level.
2. To improve the accuracy of the HAR system, bidirectional gated recurrent units (Bi-GRU) have been combined with bidirectional long short term memory (Bi-LSTM) classifiers using Dempster-Shafer theory (DST).

This paper is organized as follows: Section 2 discusses the literature on HAR. An overview of datasets is provided in Section 3. In Section 4, the proposed HAR model is described. A discussion of the experimental results is provided in Section 5. Section 6 concludes this paper.

2 RELATED WORKS

Bokhari et al. [24] proposed a method called deep gated recurrent unit (DGRU) for non-obtrusive HAR. Further, a de-noising approach based on the empirical model decomposition (EMD) has been used, followed by a linear discriminant analysis (LDA) and a discrete wavelet transform (DWT), aimed at reducing the dimensionality and extracting features from the data. It is evident from the results of the investigation that the DGRU produces the highest level of classification accuracy. A faster networking approach was developed by Xu et al. [25]. To enhance the efficiency of the optical flow features, fusion methods of spatio-temporal features were investigated, in which the temporal and spatial information was combined to form a single feature. Further, CNN with OFF was projected in place of VGG16-network, which was used to achieve a high number of features. In terms of accuracy, the proposed method outperformed these conventional schemes. A deep neural network (DNN) for HAR has been developed by Qin et al. [26] based on several sensor data sets. The DNN encoded the time series of the sensor data as images and controlled these deformed images to preserve the important characteristics of the HAR. In addition, a deep residual network (DRN) with different layers was used to accommodate the different dataset sizes. According to the results, the DNN technique outperformed the previously demanding techniques in terms of F1-score and precision.

Explaining and reasoning with knowledge-oriented and data-driven models, Jia et al. [27] developed a hierarchical structure-oriented scheme and technique for HAR. The method consists primarily of creating a hierarchical representation of the compound action based on the semantic meaning. As a result, the hierarchical approach to symbolic analysis had been established as a useful methodology. As demonstrated by Liu et al. [28], the normalized dynamic graph convolutional network (MRDGCN) continuously updates the structure of the data until an optimal model is obtained. An optimal convolution layer was constructed to determine the structure of the data. Thus, the MRDGCN has learned higher

level sample features to improve its learning performance on the data representations.

Jung et al. [29], Sena et al. [30] and L'Yvonnet et al. [31] presented a sound recognition-oriented HAR method using recurrent neural networks (RNNs). This study collected sound data by analyzing ten groups of people who performed daily concerts in the internal environment. With the help of a Log Mel-filter bank energies technique, the features have been derived from aural data, and an RNN scheme with various layers has been trained based on the aural data. In comparison with the existing models, the RNN model provided enhanced recall scores.

As a result of the DCNNs, Sena et al. [30] derived patterns using data from a variety of chronological scales. It was appropriate to use this method as the data were presented prior to a temporal series and the derived scales provided valuable information regarding the activities carried out by the users. By using this scheme, it was possible to extract both simple and composite movement. Multitemporal and multimodal systems have been developed that outperform the previous studies using two diverse datasets. The HAR scheme presented by L'Yvonnet et al. [31] is based on the possible differences in human performance. A discrete-time Markov chains (DTMC) and a PRISM model were used to examine and express the motivating temporal reasons that pertain to the dynamic development of activities within this scheme. It was observed that the DTMC's scheme performed better compared to the other schemes.

The reviews on the HAR models are summarized in Table 1. The LDA is generally used for improving the accuracy and increasing the F1-scores [24]. The tentative scenarios, however, were not measured as a result of the study. To improve the speed and accuracy, Xu et al. [25] used the CNN method. Further, the DRN model was used to provide higher F1 values and greater accuracy than previous models [26]. To accomplish this, it is necessary to focus on datasets that are larger in size. Although it is a complex method, the HMM scheme was able to achieve high reliability, as well as improve the recognition rate despite its complexity [27]. The MRDGCN scheme has been used for increasing the accuracy of the recognition rate and for improving the recognition rate; however, it is important to investigate the local invariance as well [28]. In addition, the RCNN was found to provide greater precision and recall when compared to other methods; however, it is essential to consider a variety of classes of activity [29]. Generally, the DCNN incurs a negligible bias and condensed arrays, however, the kernel selection for the DCNN must be examined [30]. Additionally, the DTMCs provided better prediction accuracy and required a shorter processing time [31]. However, the temporal reasons characteristics are not taken into account.

3 DATASET DESCRIPTION

The proposed method is evaluated using multiple benchmark datasets, including UCF-ARG [32], UCF-101 [33], Hollywood2 [34] and HMDB51 [35].

Author	Deployed Schemes	Features	Challenges
Bokhari et al. [24]	LDA	Higher accuracy Improved F1-score.	No consideration on experimental scenarios.
Xu et al. [25]	CNN	Superior speed High exactness.	Optical flow was not produced.
Qin et al. [26]	DRN	Higher accuracy Higher F1 value.	Need spotlight on advanced datasets.
Jia et al. [27]	HMM	Highly consistent Better detection.	More multifaceted.
Liu et al. [28]	MRDGCN	Higher detection rate Improved accuracy.	Requires deliberation on local invariance.
Jung et al. [29]	RCNN	Improved precision Higher recall.	Need spotlight on diverse activity classes.
Sena et al. [30]	DCNN	Negligible bias Condensed count of array.	Require assessment on kernel election.
L'Yvonnet et al. [31]	DTMCs	Least time period Higher prediction accurateness.	Need spotlight on temporal reason characteristics.

Table 1. Study on existing HAR models

3.1 UCF-ARG Dataset

There are ten different types of human activities included in the UCF-ARG dataset, including *carrying*, *digging*, *boxing*, *jogging*, *opening-closing trunks*, *clapping*, *running*, *walking*, *throwing*, and *waving*. It is a multi-view dataset collected using aerial cameras mounted on ground cameras, helium expands, and roof cameras. The videos were recorded in high resolution throughout and were divided into three sets, namely training, testing, and validation, in a ratio of 6:3:1. In Figure 1, a few samples of the video frames from the UCF-ARG dataset are presented.

Figure 1. Video frame samples from UCF-ARG dataset

3.2 UCF-101 Dataset

There are 101 different realistic action videos in the UCF-101 dataset, which depict a variety of human activities. While the videos were being recorded, there were large variations in the camera's motion. There are five categories of data that can be found in this dataset: *human-human communication*, *human-object cooperation*, *playing instrument*, *body-motion only*, and *sports*. There is a wide range of human actions in this dataset. The collected videos have been divided into three sets: training, testing, and validation. In Figure 2, a few samples of the video frames from the UCF 101 dataset are presented.

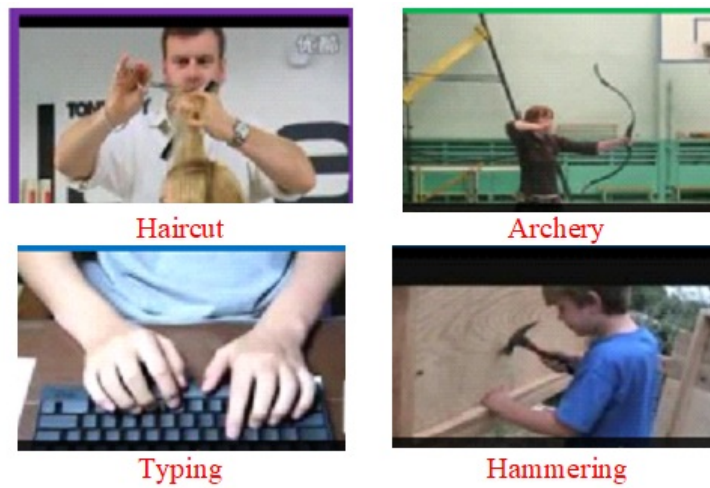


Figure 2. Video frame samples from UCF101 dataset

3.3 Hollywood2 Dataset

The Hollywood dataset contains twelve human activities. Datasets such as this one are extremely comprehensive and are considered benchmarks in the field of activity recognition. This repository contains 810 video clips in the AVI format, which were created from 69 Hollywood films. Figure 3 shows a few video frame samples from the Hollywood2 dataset.

3.4 HMDB51 Dataset

There are 51 different types of realistic action videos in the HMDB51 dataset. A variety of internet sources and digitized movies were used to collect the videos. There are five categories in this dataset: *body movements for human interaction*, *general facial actions*, *body movements with object interaction*, *general body movements*, and

Figure 3. Video frame samples from the Hollywood2 dataset

facial actions with object manipulation. The training and testing sets were divided in a 7:3 ratio between all the collected videos.

In Figure 4, a few samples of the video frames from the HMDB51 dataset are presented.

Figure 4. Video frame samples from HMDB51 dataset

4 METHODOLOGY

To classify human activity, the proposed method consists of four phases: preprocessing, segmentation, feature extraction, and classification. Figure 5 illustrates the overall architecture of the proposed work.

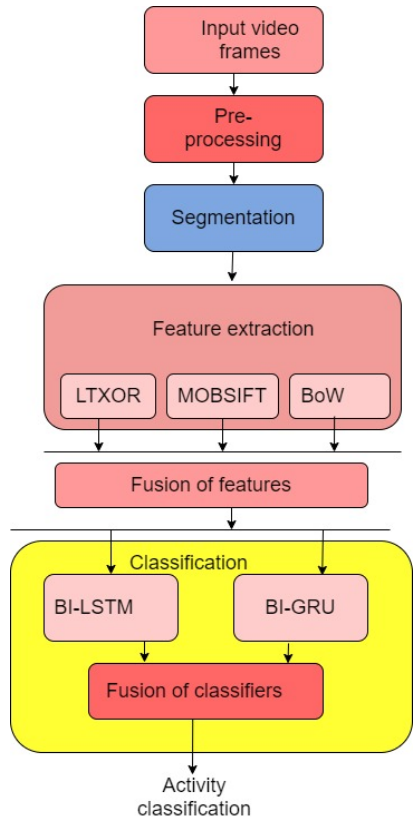


Figure 5. Overall architecture of the proposed model

4.1 Pre-Processing

As the frames have diverse backgrounds and resolutions, it becomes necessary to pre-process the images to enhance their quality.

4.1.1 Conversion of Video to Frame

Original video footage was collected which included human activities, such as *boxing*, *carrying*, *clapping*, *digging*, *jogging*, *running*, and *throwing*. Each frame of the videos was extracted using the video capture OpenCV function to extract the individual frames in the form of moving frames. These video frames are implied by *fr*, which will be used for further processing.

4.1.2 Adaptive Histogram Equalization

Adaptive histogram equalization (AHE) [36] is a digitalized image processing method that enhances the contrast of an image. It differs from the usual histogram equalization (HE) method in that the adaptive technique improves the local contrast. The HE is calculated for each division by dividing the image into separate blocks. Therefore, the AHE calculates plenty of different histograms, each of which is related to a distinct part of the image. In the different areas of the image, the contrast is improved locally and the edges are described better. However, the AHE exhibits certain noise disturbances. Therefore, in this study, a new modification is made to overcome this problem.

Conventionally, the AHE is evaluated as shown in Equation (1), where, r refers to the pixel with grey level value for new images, $P(r)$ refers to the probability density, x refers to the mean of the image and σ refers to the standard deviation of the image. As per the improved concept, the AHE is modelled as shown in Equation (2).

$$P_S(s)\,\mathrm{d}s = P_r(r)\,\mathrm{d}r, \tag{1}$$

$$P_S(s)\,\mathrm{d}s = P_r(r)\,\mathrm{d}r + G(x), \tag{2}$$

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}. \tag{3}$$

The pre-processed frames are implied as fr^{pr}.

4.2 Watershed Segmentation

When it comes to extracting regions from images, watershed segmentation [37] is a more effective and simpler method. As outlined below, there are several steps involved in the watershed segmentation process.

Step 1: Image simplification: This process helps smooth out the image and eliminates the noise interference.

Step 2: Calculation of morphological gradient image (MGI): The gray change in an image is reflected by the MGI *gr(f)* [38] that is formulated as in Equation (4), wherein s is a sign of the structuring component, Θ stands for the eroding conversion and $+$ is a sign of the dilating conversion.

$$gr(f) = (s + f) - (s\Theta f). \tag{4}$$

Step 3: This floating-point image $f^g(f)$ of activity is calculated according to Equation (5).

$$f^g(f) = \frac{gr(f) * gr(f)}{255.0}. \tag{5}$$

Step 4: Obtain the initial segmentation results similar to the watershed approach. The segmented images are implied as fr^{seg}.

4.3 Feature Extraction

From fr^{seg}, various features can be derived. The extracting features assist in identifying the most valuable characteristics and removing those that are no longer relevant. A description of the derived features is provided below.

4.3.1 LTXOR Features

The LTXOR pattern [39] uses seven different texton shapes to generate the texton images. As a preliminary step, the image is split into overlapping blocks of 2×2, which are referred to by the name B_1. The gray value positions are referred to as P, Q, R, and S for the purpose of examination. According to the shape of the texton, the subblocks are modeled as described in Equation (6).

$$Tx(Y,Z) = \begin{cases} 1, & B_1(P) = B_1(Q) \,\&\, B_1(R) \neq B_1(S), \\ 2, & B_1(Q) = B_1(S) \,\&\, B_1(P) \neq B_1(R), \\ 3, & B_1(R) = B_1(S) \,\&\, B_1(P) \neq B_1(Q), \\ 4, & B_1(P) = B_1(R) \,\&\, B_1(Q) \neq B_1(S), \\ 5, & B_1(P) = B_1(S) \,\&\, B_1(Q) \neq B_1(R), \\ 6, & B_1(Q) = B_1(R) \,\&\, B_1(P) \neq B_1(S), \\ 7, & B_1(P) = B_1(Q) \,\&\, B_1(R) = B_1(S), \\ 8, & B_1(P) \neq B_1(Q) \;\&\; B_1(R) \neq B_1(S). \end{cases} \tag{6}$$

The center of every pixel and its neighbors are collected on a texton image. After the texton image has been computed, an XOR function is applied between the center texton and its neighbors. LTXOR patterns are typically calculated as shown in Equation (7). To obtain more precise content-oriented outputs, certain modifications are made to the existing LTXOR. Based on the improved concept, the LTXOR is calculated in accordance with Equation (8), where, HM_w denotes the weighted harmonic mean formulated in Equation (9), N refers to the overall weight,

w_i refers to the weight randomly selected among 1 and 2.

$$LTXOR_{G,L} = \sum_{l=1}^{G} 2^{l-1} \times \tilde{f}_3(Tx(b_l) \otimes Tx(b_a)), \tag{7}$$

$$ILTXOR_{G,L} = \sum_{l=1}^{G} 2^{l-1} \times \frac{\tilde{f}_3(Tx(b_l) \otimes Tx(b_a))}{HM_w}, \tag{8}$$

$$HM_w = \frac{\sum_{l=1}^{n} w_i}{\sum_{l=1}^{n} \frac{w_i}{y_i}}, \tag{9}$$

$$\tilde{f}_3(y \otimes z) = \begin{cases} 1, & \text{if } y \neq z, \\ 0, & \text{else.} \end{cases} \tag{10}$$

In Equation (10), $\otimes$ points to the XOR function amid the variables, $Tx(b_a)$ points to the texton shape for the centre pixel b_a, and $Tx(b_l)$ points to the texton shape for neighbour pixel b_l. In addition, the particular image of the texton is malformed to the maps of the LTXOR within 0 to $2^{\tilde{p}-1}$. In the LTXOR calculation, it specifies the total map using the histogram construction as in Equation (11).

$$His_{LTXoR}(m) = \sum_{J=1}^{T_1} \sum_{k=1}^{T_2} \tilde{f}_2(LTXoR(\tilde{j}, \tilde{k}), \tilde{m});\ \tilde{m} \in [0, (2^{\tilde{p}} - 1)]. \tag{11}$$

The extracted LTXOR features are specified as fe^{ILT}.

4.3.2 BoW

Bag of words (BoW) [40] is most likely used as a feature representation scheme for the still images and the videos in HAR. The bag of visual words, also known as the BoW, is a symbolic scheme used to symbolize the documents for retrieval purposes. The scheme was implemented to retrieve videos and images. BoW features are indicated by the expression fe^{BOW}.

4.3.3 MoBSIFT

MoBSIFT interest point detection is similar to the working of a MoSIFT [41] detector. It behaves as a temporal expansion of the well accepted SIFT [42] model. Scale invariant feature transform (SIFT) established by Lowe is the most accepted scheme to find the feature descriptors and key points (interest points). In SIFT, the key points were the spatial interest points recognized by building the difference of Gaussian (DoG) pyramids and after that local extremes of the DoG imageries were found across the neighbouring scales. The interest point detection is mathematically

depicted as shown in Equations (12) and (13).

$$O(a, b, \sigma) = g(a, b, \sigma) * fr^{se}(a, b), \tag{12}$$

$$U(a, b, \sigma) = O(a, b, l\sigma) - O(a, b, \sigma). \tag{13}$$

In Equations (12) and (13), $O(a, b, \sigma)$ refers to the scale spacing of the input imagery $fr^{se}(a, b)$ attained by convolving it with the variable scale Gaussian, $g(a, b, \sigma)$ and $U(a, b, \sigma)$ refers to the DoG of the input imagery. The extracted MoBSIFT features are denoted by fe^{SLBT}.

4.3.4 Proposed Method for Fusion of Features

A Bi-LSTM and Bi-GRU variant of the RNN classifiers are trained using the extracted multi-features:

$$f = fe^{ILT}.fe^{BOW}.fe^{SIFT}. \tag{14}$$

4.4 Human Activity Classifcation

The Bi-GRU and Bi-LSTM classifiers have been used in the proposed work to classify human activities. The performance of the HAR was tested using both the individual classifiers as well as the combination of these classifiers. The Bi-LSTM and Bi-GRU variants of the RNN classifiers are used in this study to classify the feature vectors f from Equation (14). As the recurrently connected nodes are present in the hidden layers of the RNN, its internal states are capable of remembering inputs from several past timestamps.

4.4.1 Bi-GRU Variant of RNN Based Recognition

The Bi-GRU [40] uses exceptional gates (ut), known as reset and update gates for the declining gradient dispersion with smaller loss. The ut substitute input and forget gate of the LSTM, which depict the preservation degree of the preceding data. The proposed architecture of the Bi-GRU variant of the RNN for HAR is illustrated in Figure 6.

$$ut = \mu(W_u.(R_{t-1}, Fea_t) + f_u). \tag{15}$$

In Equation (15), μ points out the sigmoid activation function among 0 and 1, Fea_t stands for the input matrix at time step t, R_{t-1} stands for the hidden state at the prior time step $t - 1$. W_u stands for the weight matrix of ut and f_u stands for the bias matrix of ut. The rt regulates the amount of chronological data that has to be ignored which is revealed in Equation (16), wherein, W_r characterizes the weight matrix of rt and f_r symbolizes the bias matrix of rt.

$$rt = \mu(W_r.(R_{t-1}, Fea_t) + f_r). \tag{16}$$

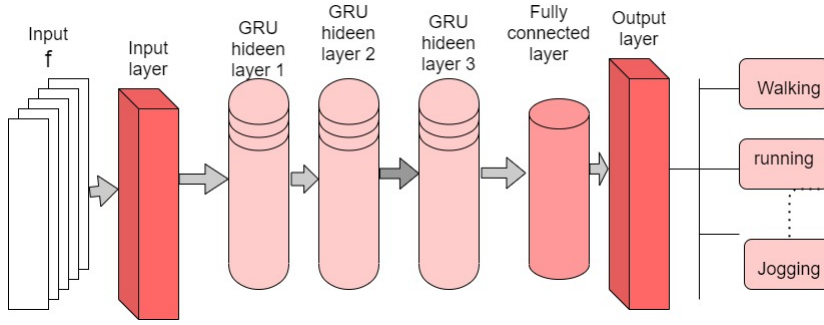


Figure 6. Proposed architecture of Bi-GRU variant of RNN for HAR in the present work

The hidden candidate state is exposed in Equation (17), wherein, tanh stands for the activation function. f_R and W_R stand for the bias matrix and weight matrix of the new cell state, * stands for the dot multiplication function. As a result, the output (R_t) implies linear disruption amid R_{t-1} and $(\tilde{R}_t)$.

$$\tilde{R}_t = \tanh(W_R.(R_{t-1} * rg, Fea_t) + f_R), \tag{17}$$

$$R_t = (1 - ut) * R_{t-1} + ut * \tilde{R}_t. \tag{18}$$

The forward and backward GRUs capture the prior and forthcoming facts of the input data. The Bi-GRU is devised as shown in Equation (19), wherein, $\overleftarrow{R_t}$ and $\overrightarrow{R_t}$ correspond to the hidden state of backward and forward GRU in that order, Ct corresponds to combining technique of the outputs at two directions.

$$Yt = Ct(\overleftarrow{R_t}, \overrightarrow{R_t}). \tag{19}$$

4.4.2 Bi-LSTM Variant of RNN Based Recognition

The Bi-LSTM classifier [39] covers a series of recurrent LSTM cells. The Bi-LSTM cells include the "forget gate, input gate, and output gate". Let, the variables be the hidden and cell state. The proposed architecture of the Bi-LSTM variant of the RNN for the HAR is illustrated in Figure 7.

(X_t, D_{t-1}, Z_{t-1}) and (Z_t, D_t) designate the input and output layer. At certain times, the output, input and forget gate implies O_t, I_t, F_t. The Bi-LSTM primarily uses F_t to sort the information. F_t is formulated as shown by Equation (20).

$$F_t = \sigma(J_{IF}X_t + L_{IF} + J_{ZF}Z_{t-1} + L_{ZF}). \tag{20}$$

In Equation (20), (J_{ZF}, L_{ZF}) and (J_{IF}, L_{IF}) points out the weight and bias constraint to map the hidden and input layers to forget that the gate and activation function is signified by σ. The input gate is exploited by the Bi-LSTM as revealed in Equations (21), (22) and (23), wherein, the (J_{ZG}, L_{ZG}) and (Z_{II}, L_{II}) imply the

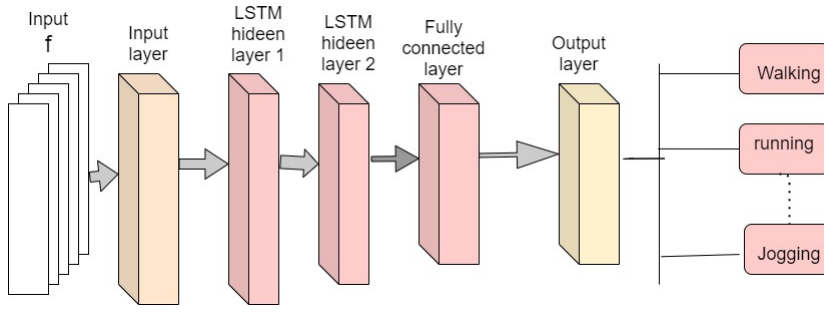


Figure 7. Proposed architecture of Bi-LSTM varient of RNN for HAR in the present work

weight and bias constraint to map the hidden and input layers to I_t.

$$G(t) = \tanh(J_{IG}X_t + L_{IG} + J_{ZG}Z_{t-1} + L_{ZG}), \tag{21}$$

$$I_t = \sigma(J_{II}X_t + L_{II} + J_{ZI}Z_{t-1} + L_{ZI}), \tag{22}$$

$$D_t = F_t D_{t-1} + I_t G_t, \tag{23}$$

$$O_t = \sigma(J_{IO}X_t + L_{IO} + J_{ZO}Z_{t-1} + L_{ZO}), \tag{24}$$

$$Z_t = O_t \tanh(D_t). \tag{25}$$

The Bi-LSTM cell obtains a hidden-layer from the output gate as shown in Equations (24) and (25), in which, (J_{ZO}, L_{ZO}) and (J_{IO}, L_{IO}) represent the weight and bias to map the hidden and the input layer to O_t.

4.4.3 Proposed Approach of Combining Bi-GRU and Bi-LSTM Variants of RNN Classifiers

The purpose of this section is to demonstrate how Bi-LSTMs and Bi-GRU variants of RNN can be combined to produce more accurate predictions from the individual class probabilities. In this process, both Bi-LSTMs and Bi-GRU variants of RNN are simultaneously trained with f and then combined with the probabilistic output of Bi-LSTMs and the probabilistic output of Bi-GRU using DST [43], as shown in Figure 8.

Fusion using DST. A DST of evidence is different from statistically based combination methods in that it is capable of representing lack of knowledge and uncertainty. This is quite important in the context of classifier combinations, since each classifier generally possesses a certain degree of uncertainty related to its performance. The DST method utilizes gradient descent learning to minimize the mean square error (MSE) between the output of a training set and the target output [43]. An illustration of a DST-based method is provided below.

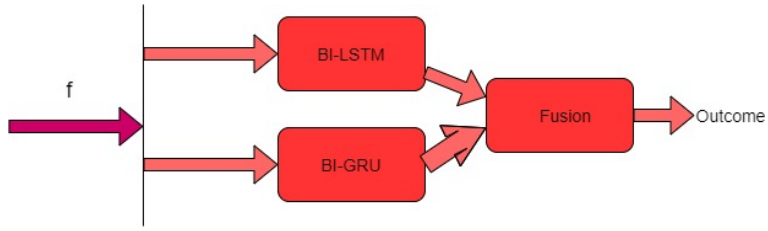


Figure 8. The proposed technique of fusing the Bi-LSTMs and Bi-GRU variants of RNN classifiers

Suppose $C = \{C_1, C_2, \ldots, C_n\}$ is a finite set of exclusive classes. There is a mass function M that is defined on the power set of C, represented by $P(C)$, which maps onto $[0, 1]$.

$$\sum M(x) = 1, x \subseteq C \text{ and } M(\phi) = 0, \tag{26}$$

$$P(C) = 2^{|C|}. \tag{27}$$

In Equation (27), $P(C)$ denotes the number of elements. The belief function *bel* is defined in terms of Equation (28), which refers to the probabilistic lower bound.

$$bel(x) = \sum M(y); \forall x \subseteq C, \text{where } y \subseteq x \text{ and } \ y \neq \phi. \tag{28}$$

In addition, the plausibility function *pf* is defined using Equation (29), which represents the probability that all the evidence does not contradict x.

$$pf(x) = \sum M(y); \forall x \subseteq C, \text{where } y \cap x \neq \phi. \tag{29}$$

$pf(x) - bel(x)$ denotes the imprecision associated with subset x of C.

Using Equation (30), let us assume that two mass functions M_1 and M_2 derived from two independent sources can be combined to form a consonant mass function *MC*.

$$MC(z) = \frac{\sum_{x \cap y = z} M_1(x) \times M_2(y)}{1 - \sum_{x \cap y = \phi} M_1(x) \times M_2(y)}. \tag{30}$$

5 EXPERIMENT RESULTS AND DISCUSSIONS

In this section, an evaluation of the proposed activity recognition approach based on different metrics including overall accuracy, precision, and recall is presented. The different benchmark datasets including the UCF-ARG, UCF-101, Hollywood2, and HMDB51 are used to evaluate the proposed approach. The performance of the proposed model is compared with the performance of the state-of-the-art models for activity recognition. It is implemented and evaluated in Python 3.9.7 on

a Windows 11 operating system (64 bit) using an Intel Core i7 processor (11^{th} generation) and a 12 GB GeForce Titan X graphics processing unit (GPU). For all the four datasets, the experiments were conducted using a stratified sample of 70 % for training and 30 % for testing. An explanation of the parameters evaluated, the results of each dataset, and a comparison with the current state-of-the-art techniques are provided below.

5.1 Hyperparameters Bi-GRU and Bi-LSTM Variants of RNN Classifiers

This study presents a Bi-LSTM model that has two hidden layers (optimal values). There are two hidden layers, the first of which processes the input sequence forward and the second of which processes it backward. A stochastic gradient descent (SGD) optimization technique has been used to determine the optimal number of hidden layers. In addition, there are 64 memory blocks that are recurrently connected in the hidden layers. A ReLU activation function has been applied to each memory block. The gates were activated using a sigmoid activation function. The softmax activation function has been used to activate the neurons in the output layer. There are three hidden layers in the Bi-GRU model, two of which process the input sequences forward and one of which processes the input sequences backward. This model uses the sigmoid functions for the control reset gate and the update gate, whereas the functions are used for the hidden state. The activation function at the hidden layer is the ReLU.

Bi-LSTM and Bi-GRU variants of the RNN accept a 128 by 128 dimensional feature vector as input. Based on the RMSprop optimizer algorithm with a learning rate of 0.0001, both networks have been optimized to minimize the categorical cross-entropy losses. A variety of the memory blocks, epochs, and batch sizes have been experimented with for each hidden layer of the Bi-LSTM and Bi-GRU. In this study, the optimal values of the various hyperparameters of the RNNs have been determined, as shown in Table 2.

5.2 Classification Results Using Bi-GRU Variant of RNN

Bi-GRU have been trained and tested on data using varying epochs, block sizes, and batch sizes. The accuracy of the proposed HAR system using the Bi-GRU variant of RNN classifiers is presented in Table 3. Based on Table 3, it can be seen that LSTMs with small batch sizes produce better recognition results. Additionally, Table 3 demonstrates that hidden layer 3 has improved recognition performance over hidden layer 1.

The performance of the HAR system using the Bi-GRU variant of the RNN classifier on UCF-101, Hollywood2, and HMDB51 dataset are given in Table 4.

Hyperparameters	**Search Space**	**Optimal Value**
Bi-LSTM hidden layer	1–3	2
Bi-GRU hidden layer	1–3	3
Memory blocks in the first Bi-LSTM hidden layer	32–64	64
Memory blocks in the second Bi-LSTM hidden layer	32–64	32
Memory blocks in the first Bi-GRU hidden layer	32–64	64
Memory blocks in the second Bi-GRU hidden layer	32–64	32
Memory blocks in the third Bi-GRU hidden layer	32–64	64
Batch size in Bi-LSTM hidden layer	20–40	30
Batch size in Bi-GRU hidden layer	10–20	10
Epochs	100–300	200
Learning rate	0.0001–0.0002	0.0001

Table 2. Optimal set of values of the various hyperparameters used in the Bi-LSTM and Bi-GRU variants of the RNN classifier

Hidden Layer	**Blocks**	**Epochs**	**Batch Size**	**Accuracy**
1	32	100	10	96.7 %
			20	96.3 %
		200	10	96.9 %
			20	96.7 %
	64	100	10	97.3 %
			20	96.5 %
		200	10	97.2 %
			20	96.8 %
2	32	100	10	97.6 %
			20	97.4 %
		200	10	98.1 %
			20	97.8 %
	64	100	10	98.1 %
			20	95.1 %
		200	10	97.4 %
			20	98.1 %
3	32	100	10	98.2 %
			20	97.8 %
		200	10	98.1 %
			20	97.9 %
	64	100	10	98.3 %
			20	98.1 %
		200	10	**98.9** %
			20	98.3 %

Table 3. Accuracy of the proposed HAR system using the Bi-GRU variant of the RNN classifier [UCF-ARG dataset]

Metric	UCF-101	Hollywood2	HMDB51
Accuracy	95.2 %	98.3 %	77.6 %
Precision	88.9 %	72.5 %	69.3 %
Recall	85.1 %	68.6 %	66.2 %

Table 4. Performance of the HAR system using the Bi-GRU variant of the RNN classifier

5.3 Classification Results Using Bi-LSTM Variant of RNN

Bi-LSTMs have been trained and tested on data using varying epochs, block sizes, and batch sizes. The accuracy of the proposed HAR system using the Bi-LSTM variant of RNN classifiers is presented in Table 5. Based on Table 5, it can be seen that hidden layer 2 has improved recognition performance over hidden layer 1.

The performance of the HAR system using the Bi-LSTM variant of the RNN classifier on UCF-101, Hollywood2, and HMDB51 dataset are given in Table 6.

5.4 Classification Results by Combining the Bi-GRU and the Bi-LSTM Variant of the RNN Classifier

Based on the proposed classifier combination discussed in Section 4.4.3, the activity recognition rates are presented in this section. On the UCF-ARG dataset, the performance of the Bi-GRU and Bi-LSTM variants of the RNN classifiers was evaluated. Using the Bi-GRU and Bi-LSTM variants of the RNN classifiers, the proposed HAR system was analysed, as shown in Table 7.

Three evaluation metrics were used to evaluate the proposed technique. The accuracy, precision, and recall for each dataset was calculated to assess the positive predictive value and sensitivity of the proposed technique. The results can be found in Figure 9.

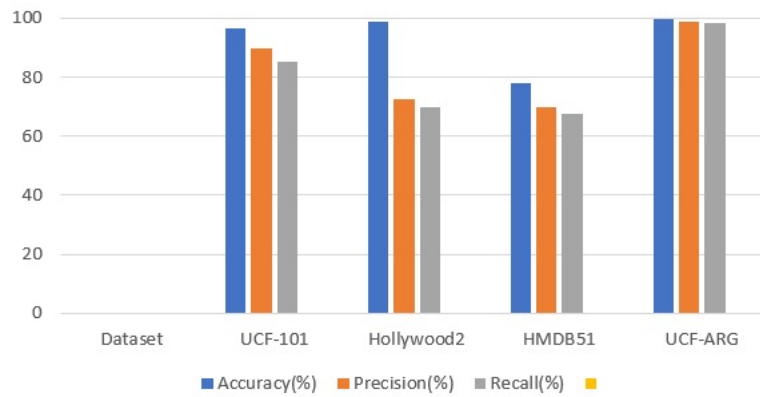


Figure 9. Evaluation of the proposed technique using accuracy, precision, and recall

Hidden Layer	Blocks	Epochs	Batch Size	Accuracy
1	32	100	20	95.7%
			30	96.1%
			40	95.6%
		200	20	95.9%
			30	96.3%
			40	95.7%
		300	20	95.5%
			30	95.9%
			40	95.4%
	64	100	20	96.3%
			30	96.5%
			40	96.1%
		200	20	96.5%
			30	96.7%
			40	96.3%
		300	20	96.1%
			30	96.3%
			40	95.9%
2	32	100	20	96.4%
			30	96.9%
			40	96.3%
		200	20	96.7%
			30	**97.4%**
			40	96.4%
		300	20	96.2%
			30	96.6%
			40	96.2%
	64	100	20	96.7%
			30	96.9%
			40	96.5%
		200	20	96.9%
			30	97.1%
			40	96.7%
		300	20	96.5%
			30	96.7%
			40	96.3%
3	32	100	20	96.2%
			30	96.7%
			40	96.1%
		200	20	96.5%
			30	97.2%
			40	96.1%
		300	20	95.9%
			30	96.3%
			40	96.0%
	64	100	20	96.4%
			30	96.6%
			40	96.2%
		200	20	96.6%
			30	96.9%
			40	96.5%
		300	20	96.2%
			30	96.4%
			40	96.0%

Table 5. Accuracy of the proposed HAR system using the Bi-LSTM variant of the RNN classifier [UCF-ARG dataset]

Metric	UCF-101	Hollywood2	HMDB51
Accuracy	94.9 %	97.9 %	77.2 %
Precision	88.1 %	72.1 %	68.6 %
Recall	84.6 %	69.2 %	67.2 %

Table 6. Performance of the HAR system using the Bi-LSTM variant of the RNN classifier

Metric	Bi-GRU	Bi-LSTM	Fusion (Bi-GRU + Bi-LSTM)
Accuracy	98.9 %	97.4 %	**99.8 %**
Precision	97.6 %	96.3 %	98.9 %
Recall	98.3 %	97.7 %	98.5 %

Table 7. Overall performance analysis of the proposed HAR system by combining the Bi-GRU and Bi-LSTM variant of the RNN classifier [UCF-ARG]

5.5 Comparison with the State-of-the-Art Results

As shown in the Table 8, some existing HAR systems available in the literature are compared to the proposed system. To evaluate the performance of the existing systems, the same datasets (UCF-ARG, UCF-101, Hollywood2 and HMDB51) were used similar to the state-of-the-art study.

6 CONCLUSION AND FUTURE WORK

In this study, an HAR scheme was developed that enabled video-to-frame conversion and AHE could be accomplished through a pre-processing step. In addition, watershed segmentation was performed, and features such as LTXOR, MoBSIFT, and BoW were extracted from the segmented images. A motion, shape, and texture feature was used to represent an activity in a selected shot. A novel deep learning model was developed to classify the human activities that combine Bi-GRU and Bi-LSTM variants of RNNs. To verify its effectiveness, the proposed system was extensively tested using different accuracy matrices for four benchmark activity recognition datasets. Due to its ability to process video streams in near real time, its low time complexity, and high detection accuracy, the system was considered suitable for industrial applications. Based on the empirical results, the proposed approach appears to be robust in the context of an HAR. As a result of this approach, it is possible to identify the activity of a single individual within a video. Further research will be conducted on individual and group activities for the HAR in the future. In addition, the multi-view datasets and complex datasets will be examined to recognize the activities.

Reference	**Classifier(s) Used**	**UCF-101**	**Hollywood2**	**HMDB51**	**UCF-ARG**
Mliki et al. [44]	LSTM	–	–	–	99.5 %
Subramanian et al. [45]	Deep genetic model	–	98.42 %	–	81.40 %
AlDahoul et al. [46]	Stochastic gradient descent	–	–	–	98 %
Burghouts et al. [47]	SVM	–	–	–	93 %
Ullah et al. [48]	LSTM	94.45 %	69.5 %	72.21 %	–
Ullah et al. [3]	DS-GRU	95.5 %	71.3 %	71.3 %	–
Xin et al. [49]	LSTM	85.3 %	63.1 %	58.2 %	–
Li et al. [50]	Bi-LSTM	94.2 %	–	70.4 %	–
Yang et al. [51]	3D-CNNs and bi-directional hierarchical LSTM	94.8 %	–	71.9 %	–
Wang et al. [52]	temporal segment network	94.2 %	–	69.4 %	–
Mahasseni et al. [53]	RLSTM	86.9 %	–	55.3 %	–
Liu et al. [54]	Hierarchical clustering	76.3 %	–	51.4 %	–
Ke et al. [55]	descriptor approaches	–	64.60 %	–	–
Lan et al. [56]	Multi-skIp Feature Stacking	89.1 %	68 %	65.4 %	–
Islam et al. [57]	SVM	–	87 %	–	–
Hou et al. [58]	FASNet, MIFS, SVM	–	78.1 %	–	–
Proposed system	**Fusion of Bi-GRU and Bi-LSTM**	**96.8 %**	**98.9 %**	**78.2 %**	**99.8 %**

Table 8. An analysis of the comparative performance with a limited number of studies already available

REFERENCES

[1] Tang, S.—Roberts, D.—Golparvar-Fard, M.: Human-Object Interaction Recognition for Automatic Construction Site Safety Inspection. Automation in Construction, Vol. 120, 2020, Art. No. 103356, doi: 10.1016/j.autcon.2020.103356.

[2] Janarthanan, R.—Doss, S.—Baskar, S.: Optimized Unsupervised Deep Learning Assisted Reconstructed Coder in the On-Nodule Wearable Sensor for Human Activity Recognition. Measurement, Vol. 164, 2020, Art. No. 108050, doi: 10.1016/j.measurement.2020.108050.

[3] Ullah, A.—Muhammad, K.—Ding, W.—Palade, V.—Haq, I. U.—Baik, S. W.: Efficient Activity Recognition Using Lightweight CNN and DS-GRU Network for Surveillance Applications. Applied Soft Computing, Vol. 103, 2021, Art. No. 107102, doi: 10.1016/j.asoc.2021.107102.

[4] Moayedi, F.—Azimifar, Z.—Boostani, R.: Human Action Recognition: Learning Sparse Basis Units from Trajectory Subspace. Applied Artificial Intelligence, Vol. 30, 2016, No. 4, pp. 297–317, doi: 10.1080/08839514.2016.1169094.

[5] Yurtman, A.—Barshan, B.: Human Activity Recognition Using Tag-Based Radio Frequency Localization. Applied Artificial Intelligence, Vol. 30, 2016, No. 2, pp. 153–179, doi: 10.1080/08839514.2016.1138787.

[6] Zhang, H.—Parker, L. E.: CoDe4D: Color-Depth Local Spatio-Temporal Features for Human Activity Recognition from RGB-D Videos. IEEE Transactions on Circuits and Systems for Video Technology, Vol. 26, 2016, No. 3, pp. 541–555, doi: 10.1109/TCSVT.2014.2376139.

[7] Yan, Y.—Ricci, E.—Liu, G.—Sebe, N.: Egocentric Daily Activity Recognition via Multitask Clustering. IEEE Transactions on Image Processing, Vol. 24, 2015, No. 10, pp. 2984–2995, doi: 10.1109/TIP.2015.2438540.

[8] Yousefzadeh, A.—Orchard, G.—Serrano-Gotarredona, T.—Linares-Barranco, B.: Active Perception with Dynamic Vision Sensors. Minimum Saccades with Optimum Recognition. IEEE Transactions on Biomedical Circuits and Systems, Vol. 12, 2018, No. 4, pp. 927–939, doi: 10.1109/TBCAS.2018.2834428.

[9] Popescu, A. C.—Mocanu, I.—Cramariuc, B.: Fusion Mechanisms for Human Activity Recognition Using Automated Machine Learning. IEEE Access, Vol. 8, 2020, pp. 143996–144014, doi: 10.1109/ACCESS.2020.3013406.

[10] Osayamwen, F.—Tapamo, J. R.: Deep Learning Class Discrimination Based on Prior Probability for Human Activity Recognition. IEEE Access, Vol. 7, 2019, pp. 14747–14756, doi: 10.1109/ACCESS.2019.2892118.

[11] Ehatisham-Ul-Haq, M.—Javed, A.—Azam, M. A.—Malik, H. M. A.—Irtaza, A.—Lee, I. H.—Mahmood, M. T.: Robust Human Activity Recognition Using Multimodal Feature-Level Fusion. IEEE Access, Vol. 7, 2019, pp. 60736–60751, doi: 10.1109/ACCESS.2019.2913393.

[12] Wu, X.—Chu, Z.—Yang, P.—Xiang, C.—Zheng, X.—Huang, W.: TW-See: Human Activity Recognition Through the Wall with Commodity Wi-Fi Devices. IEEE Transactions on Vehicular Technology, Vol. 68, 2019, No. 1, pp. 306–319, doi: 10.1109/TVT.2018.2878754.

[13] Muaaz, M.—Chelli, A.—Abdelgawwad, A. A.—Mallofré, A. C.—Pätzold, M.: WiWeHAR: Multimodal Human Activity Recognition Using Wi-Fi and Wearable Sensing Modalities. IEEE Access, Vol. 8, 2020, pp. 164453–164470, doi: 10.1109/ACCESS.2020.3022287.

[14] Wang, L.—Zhao, X.—Si, Y.—Cao, L.—Liu, Y.: Context-Associative Hierarchical Memory Model for Human Activity Recognition and Prediction. IEEE Transactions on Multimedia, Vol. 19, 2017, No. 3, pp. 646–659, doi: 10.1109/TMM.2016.2617079.

[15] Liu, W.—Zha, Z. J.—Wang, Y.—Lu, K.—Tao, D.: p-Laplacian Regularized Sparse Coding for Human Activity Recognition. IEEE Transactions on Industrial Electronics, Vol. 63, 2016, No. 8, pp. 5120–5129, doi: 10.1109/TIE.2016.2552147.

[16] Tu, Z.—Li, H.—Zhang, D.—Dauwels, J.—Li, B.—Yuan, J.: Action-Stage Emphasized Spatiotemporal VLAD for Video Action Recognition. IEEE Transactions on Image Processing, Vol. 28, 2019, No. 6, pp. 2799–2812, doi: 10.1109/TIP.2018.2890749.

[17] Chen, Z.—Zhang, L.—Cao, Z.—Guo, J.: Distilling the Knowledge from Handcrafted Features for Human Activity Recognition. IEEE Transactions on Industrial Informatics, Vol. 14, 2018, No. 10, pp. 4334–4342, doi: 10.1109/TII.2018.2789925.

[18] Yao, Y.—Liu, Y.—Liu, Z.—Chen, H.: Human Activity Recognition with Posture Tendency Descriptors on Action Snippets. IEEE Transactions on Big Data, Vol. 4, 2018, No. 4, pp. 530–541, doi: 10.1109/TBDATA.2018.2803838.

[19] Cai, L.—Liu, X.—Ding, H.—Chen, F.: Human Action Recognition Using Improved Sparse Gaussian Process Latent Variable Model and Hidden Conditional Random Filed. IEEE Access, Vol. 6, 2018, pp. 20047–20057, doi: 10.1109/ACCESS.2018.2822713.

[20] Zerrouki, N.—Harrou, F.—Sun, Y.—Houacine, A.: Vision-Based Human Action Classification Using Adaptive Boosting Algorithm. IEEE Sensors Journal, Vol. 18, 2018, No. 12, pp. 5115–5121, doi: 10.1109/JSEN.2018.2830743.

[21] Vishwakarma, D. K.—Kapoor, R.: Integrated Approach for Human Action Recognition Using Edge Spatial Distribution, Direction Pixel and R-Transform. Advanced Robotics, Vol. 29, 2015, No. 23, pp. 1553–1562, doi: 10.1080/01691864.2015.1061701.

[22] Najar, F.—Bourouis, S.—Bouguila, N.—Belghith, S.: Unsupervised Learning of Finite Full Covariance Multivariate Generalized Gaussian Mixture Models for Human Activity Recognition. Multimedia Tools and Applications, Vol. 78, 2019, No. 13, pp. 18669–18691, doi: 10.1007/s11042-018-7116-9.

[23] Uddin, M. Z.: Human Activity Recognition Using Segmented Body Part and Body Joint Features with Hidden Markov Models. Multimedia Tools and Applications, Vol. 76, 2017, No. 11, pp. 13585–13614, doi: 10.1007/s11042-016-3742-2.

[24] Bokhari, S. M.—Sohaib, S.—Khan, A. R.—Shafi, M.—Khan, A. u. R.: DGRU Based Human Activity Recognition Using Channel State Information. Measurement, Vol. 167, 2021, Art. No. 108245, doi: 10.1016/j.measurement.2020.108245.

[25] Xu, J.—Song, R.—Wei, H.—Guo, J.—Zhou, Y.—Huang, X.: A Fast Human Action Recognition Network Based on Spatio-Temporal Features. Neurocomputing,

Vol. 441, 2021, pp. 350–358, doi: 10.1016/j.neucom.2020.04.150.

[26] Qin, Z.—Zhang, Y.—Meng, S.—Qin, Z.—Choo, K. K. R.: Imaging and Fusing Time Series for Wearable Sensor-Based Human Activity Recognition. Information Fusion, Vol. 53, 2020, pp. 80–87, doi: 10.1016/j.inffus.2019.06.014.

[27] Jia, H.—Chen, S.: Integrated Data and Knowledge Driven Methodology for Human Activity Recognition. Information Sciences, Vol. 536, 2020, pp. 409–430, doi: 10.1016/j.ins.2020.03.081.

[28] Liu, W.—Fu, S.—Zhou, Y.—Zha, Z. J.—Nie, L.: Human Activity Recognition by Manifold Regularization Based Dynamic Graph Convolutional Networks. Neurocomputing, Vol. 444, 2021, pp. 217–225, doi: 10.1016/j.neucom.2019.12.150.

[29] Jung, M.—Chi, S.: Human Activity Classification Based on Sound Recognition and Residual Convolutional Neural Network. Automation in Construction, Vol. 114, 2020, Art. No. 103177, doi: 10.1016/j.autcon.2020.103177.

[30] Sena, J.—Barreto, J.—Caetano, C.—Cramer, G.—Schwartz, W. R.: Human Activity Recognition Based on Smartphone and Wearable Sensors Using Multiscale DCNN Ensemble. Neurocomputing, Vol. 444, 2021, pp. 226–243, doi: 10.1016/j.neucom.2020.04.151.

[31] L'Yvonnet, T.—De Maria, E.—Moisan, S.—Rigault, J. P.: Probabilistic Model Checking for Human Activity Recognition in Medical Serious Games. Science of Computer Programming, Vol. 206, 2021, Art. No. 102629, doi: 10.1016/j.scico.2021.102629.

[32] UCF-ARG Data Set. 2011, `http://Crcv.Ucf.Edu/Data/UCF-ARG.Php`.

[33] Soomro, K.—Zamir, A. R.—Shah, M.: UCF101: A Dataset of 101 Human Actions Classes from Videos in the Wild. CoRR, 2012, doi: 10.48550/arXiv.1212.0402.

[34] Marszalek, M.—Laptev, I.—Schmid, C.: Actions in Context. 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 2929–2936, doi: 10.1109/CVPR.2009.5206557.

[35] Kuehne, H.—Jhuang, H.—Garrote, E.—Poggio, T.—Serre, T.: HMDB: A Large Video Database for Human Motion Recognition. 2011, pp. 2556–2563, doi: 10.1109/ICCV.2011.6126543.

[36] Pizer, S. M.—Amburn, E. P.—Austin, J. D.—Cromartie, R.—Geselowitz, A.—Greer, T.—ter Haar Romeny, B.—Zimmerman, J. B.—Zuiderveld, K.: Adaptive Histogram Equalization and Its Variations. Computer Vision, Graphics, and Image Processing, Vol. 39, 1987, No. 3, pp. 355–368, doi: 10.1016/S0734-189X(87)80186-X.

[37] Li, Y.—Shi, H.—Jiao, L.—Liu, R.: Quantum Evolutionary Clustering Algorithm Based on Watershed Applied to SAR Image Segmentation. Neurocomputing, Vol. 87, 2012, pp. 90–98, doi: 10.1016/j.neucom.2012.02.008.

[38] Weickert, J.: Efficient Image Segmentation Using Partial Differential Equations and Morphology. Pattern Recognition, Vol. 34, 2001, No. 9, pp. 1813–1824, doi: 10.1016/S0031-3203(00)00109-6.

[39] Bala, A.—Kaur, T.: Local Texton XOR Patterns: A New Feature Descriptor for Content-Based Image Retrieval. Engineering Science and Technology, an International Journal, Vol. 19, 2016, No. 1, pp. 101–112, doi: 10.1016/j.jestch.2015.06.008.

[40] Agusti, P.—Traver, V. J.—Pla, F.: Bag-of-Words with Aggregated Temporal Pair-Wise Word Co-Occurrence for Human Action Recognition. Pattern Recognition Letters, Vol. 49, 2014, pp. 224–230, doi: 10.1016/j.patrec.2014.07.014.

[41] Febin, I. P.—Jayasree, K.—Joy, P. T.: Violence Detection in Videos for an Intelligent Surveillance System Using MoBSIFT and Movement Filtering Algorithm. Pattern Analysis and Applications, Vol. 23, 2020, No. 2, pp. 611–623, doi: 10.1007/s10044-019-00821-3.

[42] Lowe, D. G.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, Vol. 60, 2004, pp. 91–110, doi: 10.1023/B:VISI.0000029664.99615.94.

[43] Al-Ani, A.—Deriche, M.: A New Technique for Combining Multiple Classifiers Using the Dempster-Shafer Theory of Evidence. Journal of Artificial Intelligence Research, Vol. 17, 2002, pp. 333–361, doi: 10.1613/jair.1026.

[44] Mliki, H.—Bouhlel, F.—Hammami, M.: Human Activity Recognition from UAV-Captured Video Sequences. Pattern Recognition, Vol. 100, 2020, Art. No. 107140, doi: 10.1016/j.patcog.2019.107140.

[45] Subramanian, R. R.—Vasudevan, V.: A Deep Genetic Algorithm for Human Activity Recognition Leveraging Fog Computing Frameworks. Journal of Visual Communication and Image Representation, Vol. 77, 2021, Art. No. 103132, doi: 10.1016/j.jvcir.2021.103132.

[46] AlDahoul, N.—Sabri, A. Q. M.—Mansoor, A. M.: Real-Time Human Detection for Aerial Captured Video Sequences via Deep Models. Computational Intelligence and Neuroscience, Vol. 2018, 2018, Art. No. 1639561, doi: 10.1155/2018/1639561.

[47] Burghouts, G. J.—van Eekeren, A. W. M.—Dijk, J.: Focus-of-Attention for Human Activity Recognition from UAVs. In: Huckridge, D. A., Ebert, R. (Eds.): Electro-Optical and Infrared Systems: Technology and Applications XI. SPIE, Proceedings of SPIE, Vol. 9249, 2014, doi: 10.1117/12.2067569.

[48] Ullah, A.—Muhammad, K.—Del Ser, J.—Baik, S. W.—de Albuquerque, V. H. C.: Activity Recognition Using Temporal Optical Flow Convolutional Features and Multilayer LSTM. IEEE Transactions on Industrial Electronics, Vol. 66, 2019, No. 12, pp. 9692–9702, doi: 10.1109/TIE.2018.2881943.

[49] Xin, M.—Zhang, H.—Wang, H.—Sun, M.—Yuan, D.: ARCH: Adaptive Recurrent-Convolutional Hybrid Networks for Long-Term Action Recognition. Neurocomputing, Vol. 178, 2016, pp. 87–102, doi: 10.1016/j.neucom.2015.09.112.

[50] Li, W.—Nie, W.—Su, Y.: Human Action Recognition Based on Selected Spatio-Temporal Features via Bidirectional LSTM. IEEE Access, Vol. 6, 2018, pp. 44211–44220, doi: 10.1109/ACCESS.2018.2863943.

[51] Yang, H.—Zhang, J.—Li, S.—Luo, T.: Bi-Direction Hierarchical LSTM with Spatial-Temporal Attention for Action Recognition. Journal of Intelligent and Fuzzy Systems, Vol. 36, 2019, No. 1, pp. 775–786, doi: 10.3233/JIFS-18209.

[52] Wang, L.—Xiong, Y.—Wang, Z.—Qiao, Y.—Lin, D.—Tang, X.—Van Gool, L.: Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (Eds.): Com-

puter Vision – ECCV 2016. Springer, Cham, Lecture Notes in Computer Science, Vol. 9912, 2016, pp. 20–36, doi: 10.1007/978-3-319-46484-8_2.

[53] Mahasseni, B.—Todorovic, S.: Regularizing Long Short Term Memory with 3D Human-Skeleton Sequences for Action Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 3054–3062, doi: 10.1109/CVPR.2016.333.

[54] Liu, A. A.—Su, Y. T.—Nie, W. Z.—Kankanhalli, M.: Hierarchical Clustering Multi-Task Learning for Joint Human Action Grouping and Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 39, 2016, pp. 102–114, doi: 10.1109/TPAMI.2016.2537337.

[55] Ke, Y.—Sukthankar, R.—Hebert, M.: Efficient Visual Event Detection Using Volumetric Features. Vol. 1, 2005, pp. 166–173, doi: 10.1109/ICCV.2005.85.

[56] Lan, Z.—Lin, M.—Li, X.—Hauptmann, A. G.—Raj, B.: Beyond Gaussian Pyramid: Multi-Skip Feature Stacking for Action Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 204–212, doi: 10.1109/CVPR.2015.7298616.

[57] Islam, N.—Faheem, Y.—Din, I. U.—Talha, M.—Guizani, M.—Khalil, M.: A Blockchain-Based Fog Computing Framework for Activity Recognition as an Application to E-Healthcare Services. Future Generation Computer Systems, Vol. 100, 2019, pp. 569–578, doi: 10.1016/j.future.2019.05.059.

[58] Hou, J.—Wu, X.—Sun, Y.—Jia, Y.: Content-Attention Representation by Factorized Action-Scene Network for Action Recognition. IEEE Transactions on Multimedia, Vol. 20, 2018, No. 6, pp. 1537–1547, doi: 10.1109/TMM.2017.2771462.

Kumari Priyanka Sinha is pursuing her Ph.D. in the National Institute of Technology Patna. She is working as Assistant Professor in the Nalanda College of Engineering, Chandi, Department of Computer Science and Engineering. Her current research interests include computer vision, artificial intelligence and machine learning.

Prabhat Kumar is Professor in Computer Science and Engineering Department at the National Institute of Technology Patna, India. He is also the Professor-in-Charge of the IT Services and Chairman of Computer & IT Committee of NIT Patna. He is a member of NWG-13 (National Working Group 13) corresponding to ITU-T Study Group 13 "Future Networks, with focus on IMT-2020, cloud computing and trusted network infrastructures". He is a former HOD, CSE Department, NIT Patna and former State Student Coordinator of Bihar for Computer Society of India. He holds his Ph.D. in computer science and his M.Tech. in information technology. He has more than 100 publications in various reputed journals and international conferences. He has served as guest editor of special issues in international journals and has also edited several books published by reputed international publishers. He is also in the reviewing panel of multiple reputed SCI indexed journals. He has chaired sessions at several international conferences held in India and abroad. He is a senior member of IEEE, professional member of ACM, life member of CSI, International Association of Engineers (IAENG), Indian Society for Technical Education (ISTE) and global member of Internet Society. His research area includes wireless sensor networks, internet of things, data science, software engineering, e-governance, etc.

Rajib Ghosh is currently working as Assistant Professor in Computer Science and Engineering Department at the National Institute of Technology (NIT) Patna, India. He has more than 20 years of experiences of teaching in different engineering colleges. He completed his Ph.D. (computer science and engineering) degree at the National Institute of Technology (NIT) Patna, India. He also holds his M.Tech. degree in information technology and B.E. degree in computer science and engineering. His broader research domains are pattern recognition, machine learning and computer vision. His research areas of interest are document analysis and recognition, object detection, object tracking, human movement tracking, video surveillance, etc. His Ph.D. work explored pattern recognition methods using machine learning techniques for recognizing online handwritten text of different Indic scripts such as Bengali, Devanagari, Telugu, Tamil, etc. He has over 30 research publications in various reputed SCI-indexed as well as SCOPUS-indexed international journals and conferences of repute. He is also the reviewer of several reputed journals indexed in SCI, SCIE and SCOPUS. He is one of General Chairs in one international conference. He has also chaired sessions at several international conferences. He is a member of IEEE, life member of IUPRAI, life member of ISTE, and nominee member of CSI. He has delivered expert talks and guest lectures at various prestigious institutes.

Computing and Informatics, Vol. 42, 2023, 993–1012, doi: 10.31577/cai_2023_4_993

DEEP LEARNING BASED MISOGYNISTIC BANGLA TEXT IDENTIFICATION FROM SOCIAL MEDIA

Sarif Sultan Saruar Jahan, Raqeebir Rab, Peom Dutta,
Hossain Muhammad Mahdi Hassan Khan,
Muhammad Shahariar Karim Badhon

Ahsanullah University of Science and Technology
Department of Computer Science and Engineering
Dhaka, Bangladesh
e-mail: sjalim71@gmail.com, raqeebir.cse@aust.edu, {peomd04, mahdihassankhan10, sms.badhon}@gmail.com

Sumaiya Binte Hassan

The University of British Columbia
UBC Brain, Attention, and Reality Lab
Vancouver, Canada
e-mail: sumaiya.b.hassan@gmail.com

Ashikur Rahman

Bangladesh University of Engineering and Technology
Department of Computer Science and Engineering
Dhaka, Bangladesh
e-mail: ashikur@cse.buet.ac.bd

Abstract. Misogyny is characterized by hostility, hatred, aversion, intimidation, and violence against women. With the rise of social media, it has become one of the most convenient platforms for expressing woman-hating speech. As a result, misogyny is gaining appeal and societal standards are being violated. With millions of Bangladeshi Facebook users, misogyny is growing increasingly prevalent in Bangla

as well. In this paper, we have proposed automatically identifying misogynistic content in Bangla on social media platforms in order to evaluate the problem's challenges. As there is no existing Bangla dataset for analyzing misogynistic text, we generated our own. We have applied various deep-learning algorithms to improve the classification of misogynistic text categories. LSTM and RNN models are used for designing the model architecture in deep learning. Models are evaluated using the confusion matrix, accuracy, and f1-scores. The results indicate that LSTM outperforms RNN in terms of accuracy by 67 %.

Keywords: Misogyny, deep learning, LSTM, RNN, BERT, feature selection, natural language processing

1 INTRODUCTION

Misogyny is the hate or prejudice against women that can be linguistically manifested in numerous ways, ranging from less aggressive behaviors like social exclusion and discrimination to more dangerous expressions related to threats of violence and sexual objectification. In recent years, misogyny has increased exponentially due to the widespread global use of social media platforms such as Twitter, Facebook, Instagram, and YouTube. Misogyny appears in different forms in our society, causing incalculable harm to girls and women. In this case, social media may have been a method to guarantee free speech but social media sites must now monitor and prohibit abusive content to safeguard their users. The proliferation of misogynistic content online causes an increase in social misbehavior, promoting and instigating actual hate crimes. It creates an association between the rise of misogynistic conduct online and the number of rapes in the United States [1]. Currently, the detection of women-targeted harassment in social networks is receiving increasing attention.

Misogynistic text classification is a widely researched topic, with the majority of research conducted in European languages such as English, French, Italian, etc. It is rarely studied in the context of the Bengali language, despite the fact that it is widely spoken. Bangladesh has a population of 168.7 million, and 31.5 % of the population utilizes the internet, which translates to around 58.7 million internet users by 2022 [2]. The annual growth rate of active social media users is 25 percent (9 million) [3]. As social media platforms grow in size, misogynistic behavior is increasingly reflected in the Bangla language on these sites. In Bangladesh, almost 75 % of females use social media multiple times daily, compared to 64 % of males which is a substantial proportion [4]. While new opportunities for women's self-expression have emerged, misogyny manifests as a categorization of the feminine gender. 73 % of female internet users are victims of various cyber crimes [5]. 30 % are unfamiliar with the methods for getting assistance [3]. The psychological effects of online hateful speech extend beyond the victims to the readers as well. Due to

these forms of social media activities, negative affect disorders, loneliness, anxiety, depression, suicidal ideation, and somatic symptoms are prevalent among female social media users. Misogynistic texts can have devastating societal and personal consequences. These manifestations of misogyny are a pertinent social issue that has been explored in the scientific literature during the past several years.

In this paper, we identified the label of a text based on social media comments and classified it into one of the three categories: Stereotype and Objectification, Dominance, Derailing, Sexual Harassment, and Discredit.

We have proposed deep learning-based methods for building a multi-class analyzer, where deep Learning algorithms such as RNN, LSTM, and BERT word embeddings are employed to identify misogynistic text. In recent years, deep learning techniques have performed exceptionally well because they do not require predefined characteristics; rather, they acquire knowledge from the dataset itself. We analyze the performance of our technique on a dataset containing Bengali-language comments from several social media platforms. After thorough experiments, the RNN model exhibits 55 % accuracy and LSTM with 67 % accuracy. The following is a summary of the primary contributions of this work:

1. Construct a dataset of misogynistic text in the Bangla language for the first time.
2. Approach the automatic detection of misogyny against women using a deep-learning approach; and
3. Evaluate the performance of our technique on a dataset consisting of Bengali-language comments from multiple social media sites.

The remaining paper is organized as follows: Section 2 is a summary of the relevant work. The dataset is described in Section 3. Section 4 discusses the methodology for detecting misogyny against women. Section 5 includes the experiments conducted and analyzes the results collected. Section 6 provides a discussion of the analysis and its conclusions.

2 RELATED WORKS

There is not much work that exists on misogynistic text in the Bengali language. In contrast, English, Italian, and other languages have a substantial amount of work in this field. The majority of work is concentrated on the dataset based on Twitter. For purposes of detection, machine learning techniques such as Logistic Regression, Support Vector Machine (SVM), and Naive Bayes classifier have dominated. These classifiers are trained alongside the TF-IDF word vectorization technique and integrated with the models [2]. In addition, other approaches utilized Bag and sequences of words, Characters n-grams, and Lexicons for classification. [6] utilized three data sets collected from Twitter. The SVM Model has been implemented in TF-IDF. The data has been preprocessed with Natural Language

Toolkit (NLTK)[1]. Information Gain SVM was used to calculate the weights of lexical characteristics in order to detect misogynistic tweets. Bakarov [7] approach is designed to identify misogynistic text collected from Twitter. Their system is based on a vector space model of character n-grams and a supervised gradient-boosting classifier. Another study by Frenda et al. [8] also involved two subtasks: Misogyny Identification and Misogynistic Behavior and Target Classification to identify misogynistic content on Twitter in English and Italian. They applied lexica modeling to enrich the dictionary and used an SVM classifier with RBF for each language. The paper of Ahluwalia et al. [9] collected 5 000 tweets from Twitter and classified tweets as misogynistic or not using NLTK for tokenization. Feature extraction involved Bag of Words. They applied different Machine Learning, Deep Learning, and Ensemble Learning models for classification. The best result was obtained through Deep Learning. On the other contrary, Alawneh et al. [10] collected 4 000 labeled data from "maps.safecity" categorized into Ogling, Commenting, and Groping. Data pre-processing involved tokenization, stemming, and lemmatization. Tf-Idf was used for feature extraction, and eight classifiers, including Random Forest, Multinomial NB, SVS, Linear SVC, SGD, Bernoulli NB, DT, and K-Neighbors, were evaluated. The proposed model achieved 81 % accuracy using the SGD classifier.

The current study has also focused on Deep Learning approaches for misogynistic text detection. Ordered Neurons LSTM with XLM-RoBERTa was proposed by Ou and Li [11] for hate speech detection using a dataset that included 6 839 tweets. The K-max pooling and convolution neural network was constructed using the pre-trained multilingual model XLMRoBERTa. A linear decision function is applied following the addition of an Ordered Neurons LSTM (ONLSTM) to the prior representation. In the multilingual environment, Datta et al. [12] conducted a detection study in three languages – English, Hindi, and Bangla – using a dataset of 18 000 labeled instances categorized into Overtly aggressive, Covertly aggressive, and not aggressive. Throughout the classification process, different feature models were employed for each language, and Tf-Idf was always utilized for feature extraction. The XGBoost Classifier was utilized for English Text Classification, whereas the Gradient Boosting Classifier (GBC) was employed for Bangla and Hindi Text Classification. On the English dataset, the model achieved 58 % accuracy, on the Hindi dataset, 62.08 % accuracy, and on the Bangla dataset, 59.76 % accuracy. Yet, deeper learning could produce superior results.

Chakraborty and Seddiqui [13] employed Multinomial Naive Bayes (MNB), Support Vector Machine (SVM), and CNN-LSTM as classification algorithms in a balanced dataset of 5644 instances where 50 % were labeled as 'threat and abuse' and the rest as 'No'. The SVM classifier showed the most consistent performance with an accuracy of 78 %, which was the highest achieved among the models used. Fersini et al. [14] performed a classification task to predict Misogyny and Aggressiveness in a dataset of 4 000 samples. Pre-processing was done using Word to Vector and

[1] https://www.nltk.org/

feature extraction using Tf-Idf. Three machine learning models were used, a Shallow Model, a Convolutional Neural Network, and Fine-tuning of the Pre-trained model (a multilingual strategy using BERT). The best results were obtained using the Fine-tuning of the Pre-trained model with BERT.

The majority of relevant NLP research focuses on classifying misogynistic text in tweets written in English and other languages. Misigonystic Bengali text within social media has not been explored much. We analyzed misogynistic Bengali text in social media and developed a data set of misogynistic Bengali comments as a baseline for our research.

3 DATASET

There was no Bengali dataset for identifying misogynistic text. Thereby, we created a whole new dataset for our research. The most well-known social media platforms, Facebook, Instagram, TikTok, and YouTube public posts have been used for collecting the data. We focused on how individuals acted and thought about women, as seen by their comments. We rely heavily on the opinions of individuals associated with misogyny. To differentiate between misogynistic and non-misogynistic texts, we attempted to select non-misogynistic data that was largely linked with women.

3.1 Data Acquisition

We have collected approximately *4 000* raw data from Facebook, Instagram, TikTok and YouTube. The 55 k comments on public posts from these platforms are key source of our data. Initially, we labeled the text manually. To validate these labels we have conducted a survey. According to the survey results, we have amended our dataset to include the previously mislabeled texts. Moreover, a sociologist has validated this dataset. Furthermore, as we intended to detect these texts using deep learning models the size of the dataset was not enough. To overcome this issue we have augmented the dataset using some pre-trained BERT models. Finally, in total the dataset size is *15 k*. Some of the post links can be found here[2]. Figure 1 shows some of survey report.

3.2 Data Cleaning

Data cleaning is a crucial stage in our process. Several comments contain misspellings and a combination of languages, including Bangla, English, and Banglish. All of them were manually corrected. Table 1 shows some examples of data cleaning.

[2] `https://shorturl.at/jwNUZ`

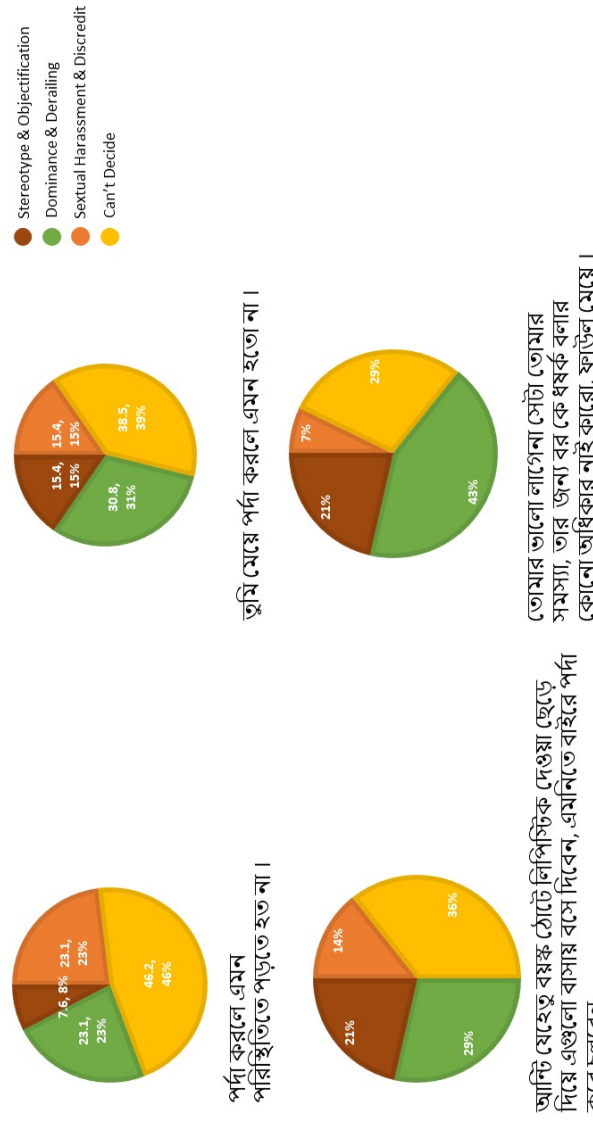


Figure 1. Sample survey report

Raw Text	Processed clean Text
যোবন তমার লাল টমাটু	যৌবন তোমার লাল টমেটো
তুই জানোস rape e মেয়েরা আসলে অনেক মজা পায়	তুই জানোস ধর্ষণে মেয়েরা আসলে অনেক মজা পায়
nari voge er jonno	নারী ভোগের জন্য

Table 1. Preview of clean processed text

3.3 Annotation Process

Depending on the misogynistic text, we grouped the misogynistic texts into three categories: stereotype and objectification, dominance, derailment, sexual harassment and the threat of violence, and discredit. Anzovino et al. [15] defined seven categories of misogynistic texts. The Bengali language contains all seven forms of misogynistic data, but some of the definitions are inconsistent with Bengali text. Hence, we have merged several categories in the Bengali language context. After examining the entire of our dataset's text we have ***three*** target classes for misogynistic texts and one for non-misogynistic texts. These categories are addressed in the next section in perspective of the considerations we have made based on the definitions mentioned by Anzovino et al. [15].

Stereotype and Objectification: Stereotypes consist of preconceived conceptions and unjustified assumptions about women, in addition to the rigid and unsophisticated image or depiction of a woman. Objectification is the act of comparing women's physical characteristics to a limited set of criteria and preconceived assumptions regarding a woman's competence for a particular task.

Example: ঘরের লক্ষীদের ঘরেই শোভা পায়, বাসে না (Female are suitable at home, not in a bus.)

Dominance and Derailing: This category consists of objects that no one has the authority to impose on women, the belief that men are superior to women, and the emphasis on gender inequity. The appearance of women is one of the primary factors upon which people form their opinions. The term dominance refers to any expression of a dominant attitude. Otherwise, we would have considered this content objectifying and stereotypical. Dominance is described as the attempt to alter a woman's words to make them more acceptable to men or the defense of any abusive behavior toward a woman. Derailment is the denial of manly accountability.

Example: তারা বের করে চলবেন আর আমরা দেখলে (They will walk out and we will see the rape.)

Sexual Harassment, Threat of Violence and Discredit: Texts that sexually insult women, solicitations for sexual favors, sexually explicit harassment, and the classification of certain activities as sexual advances are all examples of sexual harassment. The intent to physically dominate women through intimidation

is a violent threat. Women-directed abuses without a larger purpose are considered disrespectful.

Example: সুন্দরী এবং রূপবতী মেয়েরা আসলে মাগি, (Beautiful and beautiful girls are actually witches, whores and harlots.)

3.4 Data Validation

We have followed three steps in validation:

1. Comprehensive recheck of the entire annotated dataset according to the definition mentioned by Anzovino et al. [15].
2. Conducted survey on the dataset with men and women of varying ages. We updated the entire dataset based on the results of the survey.
3. We verified our level data by Sumaiya Binte Hassan, UBC Brain, Attention and Reality Lab, University of British Colombia, Vancouver, Canada.

3.5 Data Augmentation

Our data collection is quite limited for training deep learning models. Deep learning techniques require large amounts of information to enhance accuracy. Finding raw misogynistic text in Bangla is similarly challenging. Hence, one technique to increase the amount of data is to artificially add data to our dataset. Data augmentation is a well-known method for producing synthetic data from an existing dataset. In this case, Python libraries have been used. Pretrained BERT models are utilized for the purpose of augmentation. The Code snippet of data augmentation is provided[3].

Three Bengali pre-trained models were used to augment the text. Those are:

- banglabert [16],
- bangla-bert-base [17],
- sahajBERT[4].

3.6 Data Distribution

As shown in Figures 3 and 4, we separated our dataset into two parts: a binary dataset including misogynistic and non-misogynistic text. Another is a multi-class dataset in which text was categorized according to stereotype and objectification, dominance, derailment, sexual harassment and threat of violence, and discredit. The main corpus distribution is shown in Figure 2.

[3] https://github.com/sjalim/ColabCode/blob/main/text_augmentation.ipynb
[4] https://huggingface.co/neuropark/sahajBERT

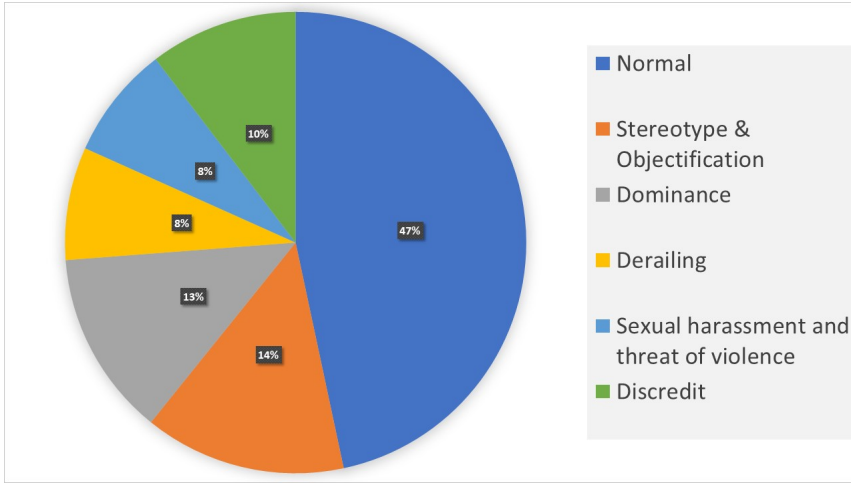


Figure 2. Main corpus distribution

4 METHODOLOGY

The purpose of this research is to identify the three types of misogynistic texts directed against women. We utilized Deep Learning techniques for classification. RNN and LSTM models are employed in the design of the neural network. Whereas the dataset was built from scratch, data pre-processing is required to obtain cleaner data. The BERT embeddings pre-trained model was used for contextual text com-

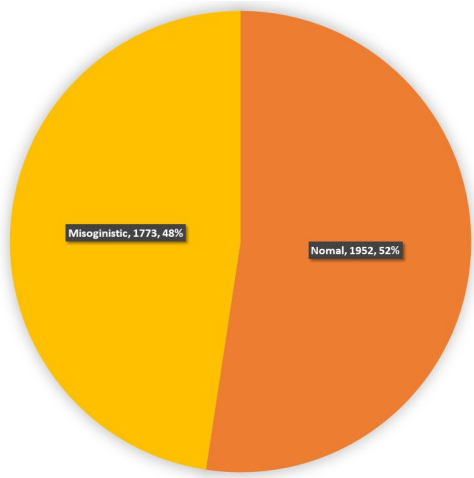


Figure 3. Binary corpus distribution

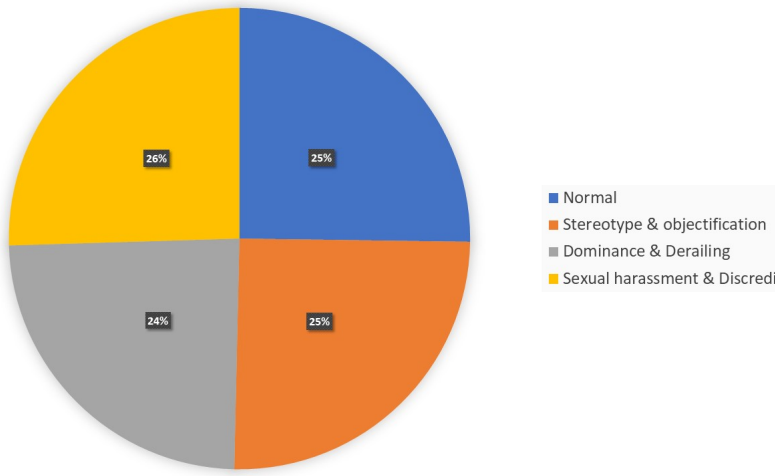


Figure 4. Multi corpus distribution

prehension in the models. Optimizers and activation functions also played a crucial role in terms of fine-tuning. The overview working procedure of our model is depicted in Figure 5.

4.1 Data Pre-Processing

Our information was gathered from several social media sites, thus it contains characters, numbers, and text fragments that can be categorized as noise and interfered with text processing. We have applied a few standard preprocessing approaches to reduce the noise.

4.1.1 POS Tagger

Sentences contain numerous non-dominant parts of speech, including conjunctions, interjections, prepositions, nouns, and pronouns, among others. When the corpus grows, the entire form of text might make text analysis cumbersome. We employed Taggers for Parts of Speech (POS tags) to identify the lexical terms included in text[5].

Examples of different types of lexical terms, their tags from Bangla Text is given below:

- Noun: মিনা, রিনা;
- Interjection: হায়, ওহ, ওঃ, ওমা;
- Pronoun: আমি, তুমি, তোমার, তার;

[5] https://bnlp.readthedocs.io/en/latest/

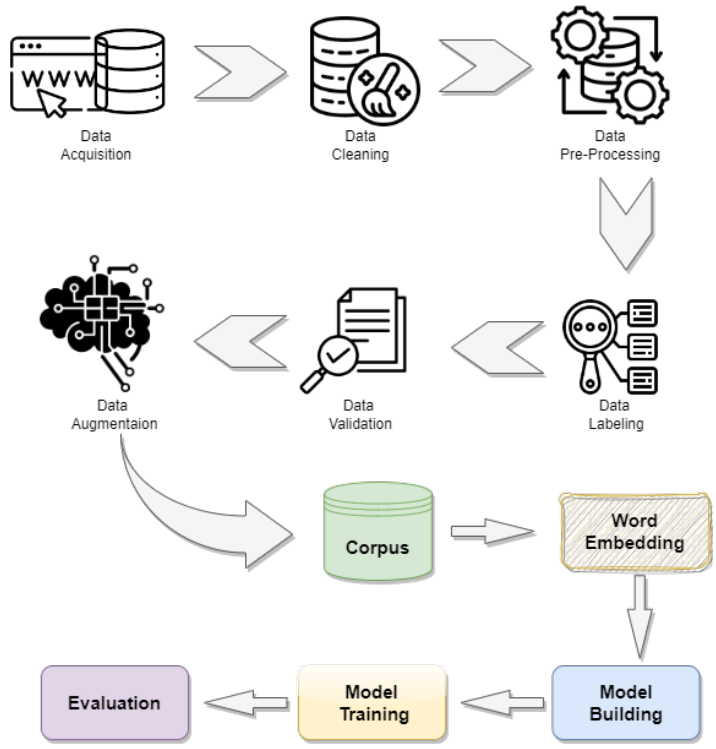


Figure 5. Work-flow diagram for the proposed model

- Preposition: দ্বারা, দিয়ে, তবে;
- Conjunction: ও, আর, এবং.

4.1.2 Removing Numbers, Punctuation, and Emoji

Numbers, punctuation, and emoji for example ১৩, 54, : ;, ☺, 🚀 are irrelevant to our work. They have no contextual significance in texts that are misogynistic. Usually, they are regarded as noise in text. The removal of numerals, punctuation, and emoji improves the performance of our model.

4.2 Word Embeddings

To extract the semantic alignment of word vectors, we have employed word embedding. For word embedding, BERT-trained models were utilized. There are numerous available strategies for feature extraction. Nonetheless, word embedding is a highly effective method for text comprehension. We have word vectorization techniques such as TF-IDF, BOW, etc. based on the frequency of words. The context of a sen-

tence is lost when its frequency is measured. On the other hand, BERT extracted contextualized word embeddings via transfer learning [18]. We have transformed every text using sbert[6] sentence transformer [19].

On our dataset, three pre-trained BERT embedding models such as BanglaBERT, Bangla BERT Base, and SahajBERT were evaluated for generating word embeddings in Bengali text. These three models performed the best with the proposed model architecture for deep learning.

4.2.1 BanglaBERT

This model gives 768 embeddings for every sentence. They have trained 27.5 GB bangla data from different websites [16].

4.2.2 Bangla Bert Base

This model gives 768 embeddings for every sentence. They have used Wikipedia Dump Dataset and OSCAR[7] Bengali dataset to train this pre-trained model [17].

4.2.3 SahajBERT

This model gives 1024 embeddings for every sentence. Collaboratively pre-trained model[8] on Bengali language using masked language modeling (MLM) and Sentence Order Prediction (SOP) objectives. They have also used Wikipedia and OSCAR datasets.

4.3 Model Architecture

Since we work with texts, sequence ordering is crucial in this case. We designed the neural network architecture for text classification using RNN and LSTM models.

RNN can learn from sequential data. In the core of its architecture, however, we frequently observe hidden layers in which entire learned knowledge is transferred from one layer to another. This can lead to issues. Because one layer may not require the knowledge of another layer to comprehend the intricate context of a sentence. In contrast, if part of the layers did not learn anything from the text, they would send a gradient of zero to the following layer, resulting in a problem with vanishing gradients. So initially, we utilized the RNN layer for model development. We have employed the LSTM model to further improve this model.

LSTM model with cell state resolved the issue. It can calculate the quantity of information that must be sent together with the cell's status. Other than this, the

[6] https://www.sbert.net/

[7] https://oscar-project.org/

[8] https://huggingface.co/neuropark/sahajBERT

top-level architecture is comparable to the RNN model. In addition to the RNN and LSTM models, our design consists of five (5) linear layers. Using ReLU and Softmax activation routines. Our model architecture is shown in Figure 6.

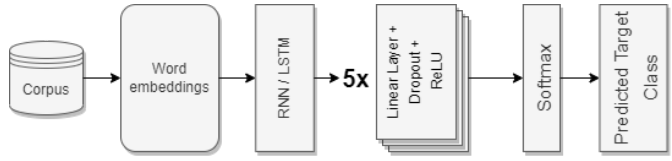


Figure 6. Model architecture

In addition to the architecture, we have experimented with various optimizers, including Adam, RAdam, and NAdam. Despite the fact that we are dealing with a multi-class issue we utilized Cross Entropy Loss for loss computation. As PyTorch can handle sparse target labels, it was utilized to implement the entire model architecture. We used a learning rate of 0.001, a batch size of 10, and 10 epochs to train the model. Around one million parameters were included in the model. Table 2 shows RNN Uni-Directional and RNN Bi-directional models parameter values. We have Table 3 for LSTM Uni-directional and LSTM Bi-directional. Finally, we have Table 4 for parameter values.

Layer	**Text Size**	**Weight Matrix Dimensions**	**Bias Vector Dimensions**
Embedding	Vocab Size×Input Size	Vocab Size×Input Size	Input Size
RNN	Hidden Size	Input Size × Hidden Size	Number of Layers × Hidden Size
Fully Connected 1	512	Hidden Size × 512	512
Fully Connected 2	256	512 × 256	256
Fully Connected 3	128	256 × 128	128
Fully Connected 4	64	128 × 64	64
Fully Connected 5	Number of Classes	64×Number of Classes	Number of Classes

Table 2. RNN parameters

5 EXPERIMENTAL RESULTS AND DISCUSSION

Our research is carried out using RNN and LSTM deep learning techniques for three different pre-trained BERT word embedding models. Accuracy, F1 score, and Cohen Kappa were used to analyze our findings. Our dataset has been divided into training

Layer	**Text Size**	**Weight Matrix Dimensions**	**Bias Vector Dimensions**
Embedding	Vocab Size x Input Size	Vocab Size × Input Size	Input Size
LSTM	Hidden Size	(Input Size + Hidden Size) × 4 × Hidden Size	4 × Number of Layers × Hidden Size
Fully Connected 1	512	Hidden Size × 512	512
Fully Connected 2	128	512 × 128	128
Fully Connected 3	Number of Classes	128 × Number of Classes	Number of Classes

Table 3. LSTM parameters

Parameter Name	**Value**
Vocab Size	15 000
Hidden Size	768
Input Size	768
Number of Layers	2
Number of Classes	2 (Binary)/4(Multi-Class)

Table 4. Parameters value

and testing portions as indicated in Table 5. 20 % of our corpus is used for testing, while 80 % is used for training.

We have conducted several experiments with our corpus. We followed some of the key steps such as a) Model Parameter tuning, b) Dataset reforming, c) Analysis results, etc. Regarding these steps' outcomes, we have made necessary changes to our model as well as to the corpus. The best possible result using our models has been portrayed by Table 6 and Table 7. Here, we see Table 6 is for Binary target class corpus and Table 7 is Multi-class.

Purpose	Count (Binary Class)	Count (Multi-Class)
Training Data	11 894	7 638
Testing Data	2 974	1 910

Table 5. Data distribution

From all the experiments it can be said that our corpus is performing well with the LSTM model where as the RNN model is lacking a bit comparatively. But if we see depending on different pre-trained BERT models the output differs. Those pre-trained models were trained with the different datasets we have mentioned in the above dataset creation section.

We have observed that all three of the BERT pre-trained model **BanglaBert-Base** performed very well. To evaluate our model's result several evaluation matrices

have been used such as Confusion Matrix, Accuracy, F1-Score (macro, weighted), and Cohen Kappa score.

Confusion Matrix gives us a proper understanding of model predicted result. Here, in Figure 7 matrix diagonally we see greater values compared to the second diagonal. A few data were unable to predict by our models. Though we have used the same corpus to train, the results are better with the LSTM model.

On the other side, we have Figure 8 matrix that shows the multi-class performance by our model with our corpus. Similarly, we see the diagonally metric is far better than other cells. The LSTM model performed better than the RNN model. Here the pre-trained model used in both is the same, but, depending on the model, the result defers.

$$\kappa = (p_o - p_e)/(1 - p_o). \tag{1}$$

Three optimizers were used, and among those, **Adam** shows huge potential. All the experiments using Adam perform phenomenally.

Cohen Kappa evaluation metric is used which is a subtle version of accuracy. Equation (1) is the formula to calculate the result, where p_e is the predicted agreement when both annotators issue labels randomly, and p_o is the empirical probability of agreement on the label assigned to each sample calculated using an empirical per-annotator prior over the class labels. Micro, weighted F1-Score has been used to evaluate the model. As we are working with multi-class classification problems macro and weighted value gives a better understanding of classwise performance.

Pre-trained	Model + Optimizer	Accuracy	F1-Score	Cohen Kappa
BanglaBert	RNN + Radam	77.17 %	73.11 %	53.59 %
	LSTM + Nadam	77.37 %	74.88 %	54.34 %
BanglaBertBase	RNN + Radam	77.27 %	74.06 %	53.97 %
	LSTM + Adam	**82.59 %**	**81.17 %**	**64.98 %**
SahajBert	RNN + Adam	78.21 %	76.15 %	56.11 %
	LSTM + Nadam	79.62 %	76.77 %	58.75 %

Table 6. Result of Binary class target

Pre-trained	Model + Optimizer	Accuracy	F1-Score (Macro)	F1-Score (Weighted)	Cohen Kappa
BanglaBert	RNN + NAdam	55.28 %	54.34 %	54.82 %	40.27 %
	LSTM + Adam	54.71 %	53.90 %	54.22 %	38.78 %
BanglaBertBase	RNN + RAdam	55.23 %	54.97 %	55.76 %	40.28 %
	LSTM + Adam	**67.27 %**	**66.91 %**	**67.11 %**	**56.26 %**
SahajBert	RNN + Adam	54.97 %	54.26 %	54.80 %	39.78 %
	LSTM + NAdam	56.38 %	55.82 %	55.74 %	41.76 %

Table 7. Result of Multi-class target

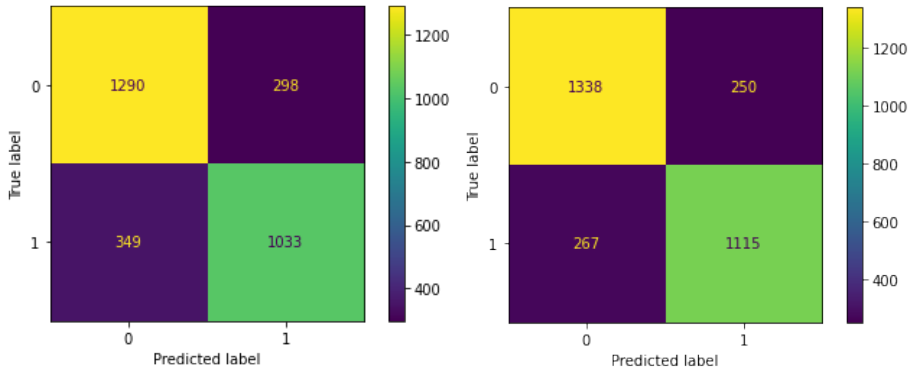


Figure 7. Left: SahajBert (RNN), Right: BanglaBertBase (LSTM) Confusion Matrix

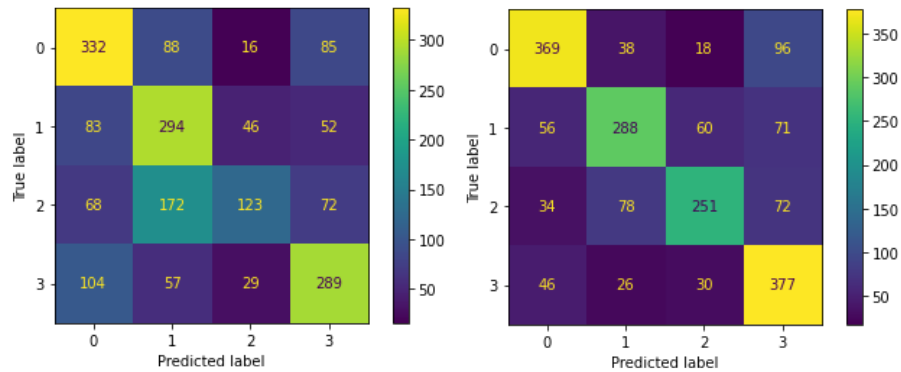


Figure 8. Left: BanglaBertBase (RNN), Right: BanglaBertBase (LSTM) Confusion Matrix

6 CONCLUSION AND FUTURE WORK

We have represented the dataset creation process on misogynistic text as well as proposed deep learning model architecture with LSTM and RNN models. Binary class and Multi-class targets both have been portrayed in this paper. The binary class target accomplished the highest accuracy of 90 % with the LSTM model. On the other hand, multi-class targets accomplished 94 % accuracy with the LSTM model. We have shown the performance of pre-trained BERT models with our dataset. As our dataset size is not enough for deep learning work at its best, we intend to increase the dataset. Thus our model is going to perform much better than now. Also, we will build a Google Chrome add-on to integrate our Misogynistic Text detection system and deploy the system as soon as possible.

REFERENCES

[1] Fulper, R.—Ciampaglia, G. L.—Ferrara, E.—Ahn, Y. Y.—Flammini, A.—Menczer, F.—Lewis, B.—Rowe, K.: Misogynistic Language on Twitter and Sexual Violence. Proceedings of the ACM Web Science Workshop on Computational Approaches to Social Modeling (ChASM'14), 2014.

[2] Shushkevich, E.—Cardiff, J.: Misogyny Detection and Classification in English Tweets: The Experience of the ITT Team. In: Caselli, T., Novielli, N., Patti, V., Rosso, P. (Eds.): EVALITA Evaluation of NLP and Speech Tools for Italian. Proceedings of the Final Workshop (EVALITA 2018). Accademia University Press, Torino, Italy, 2018, pp. 182–187, doi: 10.4000/books.aaccademia.4670.

[3] Akter, F.: Cyber Violence Against Women: The Case of Bangladesh. 2018, `https://genderit.org/articles/cyber-violence-against-women-case-bangladesh`.

[4] Auxier, B.—Anderson, M.: Social Media Use in 2021. 2021, `https://www.pewresearch.org/internet/2021/04/07/social-media-use-in-2021/`.

[5] Halim, T.: 73% Women Face Cyber Crimes: Tarana. 2017, `https://www.thedailystar.net/country/73-women-face-cyber-crimes-tarana-1372849`.

[6] Frenda, S.—Ghanem, B.—Montes-y Gómez, M.—Rosso, P.: Online Hate Speech Against Women: Automatic Identification of Misogyny and Sexism on Twitter. Journal of Intelligent and Fuzzy Systems, Vol. 36, 2019, No. 5, pp. 4743–4752, doi: 10.3233/JIFS-179023.

[7] Bakarov, A.: Vector Space Models for Automatic Misogyny Identification. In: Caselli, T., Novielli, N., Patti, V., Rosso, P. (Eds.): EVALITA Evaluation of NLP and Speech Tools for Italian. Proceedings of the Final Workshop (EVALITA 2018). Accademia University Press, Torino, Italy, 2018, pp. 211–213, doi: 10.4000/books.aaccademia.4740.

[8] Frenda, S.—Ghanem, B.—Guzmán-Falcón, E.—Montes-y Gómez, M.—Villaseñor-Pineda, L.: Automatic Expansion of Lexicons for Multilingual Misogyny Detection. In: Caselli, T., Novielli, N., Patti, V., Rosso, P. (Eds.): EVALITA Evaluation of NLP and Speech Tools for Italian. Proceedings of the Final Workshop (EVALITA 2018). Accademia University Press, Torino, Italy, 2018, pp. 188–193, doi: 10.4000/books.aaccademia.4680.

[9] Ahluwalia, R.—Soni, H.—Callow, E.—Nascimento, A.—De Cock, M.: Detecting Hate Speech Against Women in English Tweets. In: Caselli, T., Novielli, N., Patti, V., Rosso, P. (Eds.): EVALITA Evaluation of NLP and Speech Tools for Italian. Proceedings of the Final Workshop (EVALITA 2018). Accademia University Press, 2018, doi: 10.4000/books.aaccademia.4698.

[10] Alawneh, E.—Al-Fawa'reh, M.—Jafar, M. T.—Al Fayoumi, M.: Sentiment Analysis-Based Sexual Harassment Detection Using Machine Learning Techniques. 2021 International Symposium on Electronics and Smart Devices (ISESD), IEEE, 2021, pp. 1–6, doi: 10.1109/ISESD53023.2021.9501725.

[11] Ou, X.—Li, H.: YNU_OXZ @ HaSpeeDe2 and AMI: XLM-RoBERTa with Ordered Neurons LSTM for Classification Task at EVALITA 2020. In: Basile, V., Croce, D., Maro, M., Passaro, L. C. (Eds.): EVALITA Evaluation of NLP and Speech Tools for

Italian. Proceedings of the Final Workshop (EVALITA 2020). Accademia University Press, Torino, Italy, 2020, pp. 102–109, doi: 10.4000/books.aaccademia.6912.

[12] Datta, A.—Si, S.—Chakraborty, U.—Naskar, S. K.: Spyder: Aggression Detection on Multilingual Tweets. Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying, European Language Resources Association (ELRA), Marseille, France, 2020, pp. 87–92, https://aclanthology.org/2020.trac-1.14.

[13] Chakraborty, P.—Seddiqui, M. H.: Threat and Abusive Language Detection on Social Media in Bengali Language. 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), IEEE, 2019, pp. 1–6, doi: 10.1109/ICASERT.2019.8934609.

[14] Fersini, E.—Nozza, D.—Rosso, P.: AMI @ EVALITA2020: Automatic Misogyny Identification. In: Basile, V., Croce, D., Maro, M., Passaro, L. C. (Eds.): EVALITA Evaluation of NLP and Speech Tools for Italian. Proceedings of the Final Workshop (EVALITA 2020). Accademia University Press, Torino, Italy, 2020, pp. 21–28, doi: 10.4000/books.aaccademia.6764.

[15] Anzovino, M.—Fersini, E.—Rosso, P.: Automatic Identification and Classification of Misogynistic Language on Twitter. In: Silberztein, M., Atigui, F., Kornyshova, E., Métais, E., Meziane, F. (Eds.): Natural Language Processing and Information Systems (NLDB 2018). Springer, Cham, Lecture Notes in Computer Science, Vol. 10859, 2018, pp. 57–64, doi: 10.1007/978-3-319-91947-8_6.

[16] Bhattacharjee, A.—Hasan, T.—Ahmad, W.—Mubasshir, K. S.—Islam, M. S.—Iqbal, A.—Rahman, M. S.—Shahriyar, R.: BanglaBERT: Language Model Pretraining and Benchmarks for Low-Resource Language Understanding Evaluation in Bangla. In: Carpuat, M., de Marneffe, M. C., Meza Ruiz, I. V. (Eds.): Findings of the Association for Computational Linguistics: NAACL 2022. 2022, pp. 1318–1327, doi: 10.18653/v1/2022.findings-naacl.98.

[17] Sarker, S.: BanglaBERT: Bengali Mask Language Model for Bengali Language Understanding. 2020, https://github.com/sagorbrur/bangla-bert.

[18] Devlin, J.—Chang, M. W.—Lee, K.—Toutanova, K.: BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. CoRR, 2018, doi: 10.48550/arXiv.1810.04805.

[19] Reimers, N.—Gurevych, I.: Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks. CoRR, 2019, doi: 10.48550/arXiv.1908.10084.

Sarif Sultan Saruar Jahan received his B.Sc. degree in computer science and engineering from the Ahsanullah University of Science and Technology (AUST), Dhaka, Bangladesh. Currently, he is working as a Software Engineer at the BJIT Group. His research interests are NLP, image processing, deep learning, and transfer learning.

Raqeebir Rab received her B.Sc. degree in science with a major in computer science from the Augustana Faculty, University of Alberta, Canada in 2004. She completed her M.Sc. in computer science at the Concordia University, Montreal, Canada in 2012. She is an Assistant Professor at the Department of Computer Science and Engineering (CSE), Ahsanuallah University of Engineering and Technology (AUST), Dhaka, Bangladesh. Her research interests include wireless multihop networks (ad hoc and sensor networks) with an emphasis on mathematical modeling, performance analysis, protocol design, and data science.

Peom Dutta received his B.Sc. degree in computer science and engineering from the Ahsanullah University of Science and Technology (AUST), Dhaka, Bangladesh. Currently, he is working as an Associate Data Analyst. His research interests include machine learning and artificial intelligence.

Hossain Muhammad Mahdi Hassan Khan received his B.Sc. degree in computer science and engineering from the Ahsanullah University of Science and Technology (AUST), Dhaka, Bangladesh, in 2022. Currently, he is working as a Software Quality Assurance Intern. His research interests are machine learning, artificial intelligence, usage of deep learning in software testing, and cyber security.

Muhammad Shahariar Karim Badhon received his B.Sc. degree in computer science and engineering from the Ahsanullah University of Science and Technology (AUST), Dhaka, Bangladesh. Currently, he is working as a trainee at B-JET (Bangladesh-Japan ICT Engineers' Training Program). His research interests are machine learning and artificial intelligence.

Sumaiya Binte Hassan received her B.Sc. degree in cognitive systems in the cognition and brain stream from the University of British Columbia, Canada. Currently, she is working as a Research Assistant (RA) in the Brain, Attention, and Reality Lab, Canada. During her time as a Research Assistant, she focused mostly on VR and survey-based studies in the domain of evolutionary psychology.

Ashikur Rahman is a Professor in the Department of Computer Science and Engineering at Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh. He holds B.Sc. and M.Sc. degrees from BUET and his Ph.D. from the University of Alberta, Canada. He has worked as a post-doctoral researcher at various universities and his research focuses on cyber-physical systems, wireless networks, machine learning, and neural networks.

Computing and Informatics, Vol. 42, 2023, 1013–1036, doi: 10.31577/cai_2023_4_1013

AN ONLINE ENSEMBLE LEARNING MODEL FOR DETECTING ATTACKS IN WIRELESS SENSOR NETWORKS

Hiba Tabbaa, Samir Ifzarne, Imad Hafidi

Laboratory of Process Engineering, Computer Science and Mathematics (LIPIM)
University Sultan Moulay Slimane
Khouribga, Morocco
e-mail: hiba.tabbaa@usms.ac.ma, sifzarne@gmail.com, i.hafidi@usms.ma

Abstract. In today's modern world, the usage of technology is unavoidable, and the rapid advances in the Internet and communication fields have resulted in the expansion of wireless sensor network (WSN) technology. However, WSN has been proven to be vulnerable to security breaches. The harsh and unattended deployment of these networks, combined with their constrained resources and the volume of data generated, introduces a major security concern. WSN applications are extremely critical, it is essential to build reliable solutions that involve fast and continuous mechanisms for online stream analysis, allowing the identification of attacks and intrusions. Our aim is to develop an intelligent and efficient intrusion detection system by applying an important machine learning concept known as ensemble learning in order to improve detection performance. Although ensemble models have been proven to be useful in offline learning, they have received less attention in streaming applications. In this paper, we examine the application of different homogeneous and heterogeneous online ensembles in sensory data analysis on a specialized WSN detection system (WSN-DS) dataset in order to classify four types of attacks: Blackhole attack, Grayhole, Flooding, and Scheduling among normal network traffic. Among the proposed novel online ensembles, both the heterogeneous ensemble consisting of an Adaptive Random Forest (ARF) combined with the Hoeffding Adaptive Tree (HAT) algorithm and the homogeneous ensemble HAT made up of 10 models achieved higher detection rates of 96.84 % and 97.2 %, respectively. The above models are efficient and effective in dealing with concept drift while taking into account WSN resource constraints.

Keywords: Wireless sensor networks, attack detection, network security, intrusion detection system, ensemble learning, online learning, streaming data

1 INTRODUCTION

Over the past decade, through the continued increasing growth of the Internet the amount of services that come along with it has influenced almost every aspect of our human being's life. Rapid technological advances in microelectronics on the one hand and wireless communication technologies on the other have resulted in the development of affordable, versatile, and ubiquitous embedded sensor systems.

The Internet of Things (IoT) is an innovation that allows communication between an extensive variety of intelligent electronic devices and sensors. Wireless Sensor Networks (WSN) are another rapidly developing technology that is employed in IoT systems. WSNs have attracted the attention of researchers and research and development departments, and it will not be an overstatement to consider this technology as one of the most researched areas in the last decade due to their ease of deployment and their wide fields of real-time applications that differ based on their own objectives and specific constraints. The areas of WSN's applications are various, including security and surveillance, home automation, health care services, flora and fauna, urban, critical military surveillance, environment monitoring, and so forth [1]. The WSN market was valued at $ 46.76 billion in 2020 and is expected to reach $ 123.93 billion by 2026, at a Compound annual growth rate (CAGR) of 17.64 % over the forecast period of 2021–2026. Therefore, the applications of the WSN network are growing on a day-to-day basis in a considerable way.

However, a WSN has several resource constraints that include a limited amount of energy, a short communication range, low bandwidth, and limited processing and storage capabilities in each node. In many applications of WSNs, Sensor Nodes (SN) are deployed in remote, hostile, and unattended locations; therefore, it is impractical to carry out maintenance on the nodes after installation. In fact, the energy consumption of the sensors plays an important role in the lifetime of the network and has become the predominant performance criterion in this field. Additionally, in such an environment, these SNs may be subject to disruptive and malicious actions that may outright damage the proper functioning of the network. Applications of WSNs require a high level of security to provide basic security requirements such as confidentiality, integrity, authentication, availability of the data traffic, and battery life of the SNs [2, 3]. Making these applications invulnerable to different types of threats and attacks such as Blackholes, Sinkholes, Greyholes and so forth. These malicious attacks all cause the network traffic to deviate from the normal traffic, for instance, by causing the interception of data sent or received by a wireless medium and, subsequently, the ability to modify and replay the data. The intruder can also inject, saturate, or damage network equipment. In critical applications, such attacks can be harmful and cause major economic and security damage.

There are different solutions that can be used to secure WSNs, such as key management, authentication, or cryptography. Notwithstanding, these solutions do not guarantee complete prevention of all existing attacks. The hardest challenge that the entire security sector faces is detecting and dealing with upcoming attacks.

However, it is well known that intrusion detection systems (IDSs) are very effective security mechanisms to monitor the network for vicious attacks or unauthorized access as a second line of defense and alert administrators on this subject [4]. To summarize, IDS is necessary to defend against WSN attacks.

The application of Machine Learning (ML) models in order to detect possible maliciousness in WSNs has largely increased in the last decade; however, the general approach in the literature still considers the analysis as an offline learning problem, where models are trained only once on historical data. Because of the rising amount of data required to uncover increasingly sophisticated attacks and the large amount of data generated in real-time that gushes through these networks on a regular basis, traditional detection systems are inadequate for detecting malicious network intrusions. The detection of attacks requires fast mechanisms for online analysis of thousands of events per second. This encourages the creation of a fast IDS for analyzing real-time network traffic and determining instantaneously whether it is normal or exposed to any type of abnormal activity.

Stream machine learning consists of providing only a single sample (or a small batch of instances) to the learning algorithm at every instant, with a very limited processing time, a finite amount of memory, and the necessity of having trained models at every scan of the streams. In addition, robust stream-based learning algorithms must be capable of detecting drifts and updating their underlying models since a shift in data distribution (concept drift) can sometimes impact these streams of data, forcing the machine learning model to learn under non-stationary conditions. And yet, individual online learning methods are generally distinguished by a reduced detection rate. The second important requirement, aside from fast IDS, that should be considered when designing any IDS scheme for WSN is that IDS must have a high accuracy and detection rate when detecting intruders [5].

Ensemble Learning (EL) approaches are based on the idea of gaining benefit from various classifiers by learning in an ensemble way. Since some classifiers may perform well for detecting a specific type of attack but show poor performance on other types. The EL works by building on the strengths of various classifiers through a combination of their results and then generating a majority vote for classification. As a result, EL leads to maximizing accuracy through a reduction in variance and avoiding over-fitting [6].

In this paper, we propose an ensemble stream-based machine learning approach for anomaly detection tailored to the WSN's characteristics.

Contributions. Our main contributions are summarized as follows:

- Evaluate the classification performance of multiple online individual algorithms, such as k-Nearest Neighbor (KNN), Support Vector Machine (SVM), Naive Bayes (NB), and so on, under WSN for malicious intrusion detection.
- Developing an investigation methodology to study the performance of different homogeneous ensemble approaches, such as Hoeffding Adaptive Tree

(HAT), and Adaptive Random Forest (ARF), along with heterogeneous ensembles based on two base-learners, such as ARF and NB.
- Examination of the ensemble performance with the existence of concept drift.

Paper Organization. The remainder of this paper is organized as follows: Section 2 presents related work in this field. Section 3 presents the proposed online intrusion detection scheme for WSN. Section 4 presents the experimental environments of our study, and Section 5 analyzes the performance evaluation of the proposed approach. Finally, conclusions and future work are drawn in Section 6.

2 RELATED WORK

In defending against malicious attacks and misapprehensions in WSNs, various intrusion detection approaches have been proposed in the literature, and they are mainly divided into anomaly detection, misuse detection, specification-based detection, and hybrid system detection [7]. Recent research has been mostly concerned with anomaly-based IDS; thus, our research focuses on this class. Anomaly-based IDS searches for both known and unknown patterns [8]. Some credible anomaly detection approaches are currently provided based on the requirements of wireless sensor networks, notably machine learning algorithms to create a classification model based on network traffic characteristics, artificial immune algorithms, clustering algorithms, and statistical learning models.

Alqahtani et al. [9] have proposed a GXGBoot model to detect minority classes of attacks based on a genetic algorithm and an extreme gradient boosting (XGBoost) classifier in highly imbalanced data traffic in wireless sensor networks. A set of experiments were conducted on the WSN-DS dataset using held-out splitting and 10 fold cross-validation techniques. 10-fold cross-validation tests achieved satisfactory results with high detection rates of 98.2 %, 92.9 %, 98.9 %, and 99.5 % for flooding, scheduling, grayhole, and blackhole attacks, respectively, in addition to 99.9 % for normal traffic.

Park et al. [10] compared a random forest (RF) classifier with an artificial neural network (ANN) algorithm for detecting the type of DoS attacks in WSNs, and it is found that the proposed RF classifier attains the best F1-Score results, which are 96 %, 99 %, 98 %, 96 % and 100 % for flooding, blackhole, grayhole, scheduling (TDMA), and normal attacks, respectively. However, the outcome of this analysis was for a limited number of instances in the testing phase, which represents approximately 25 % (94,042 instances) of the results.

Biswas and Samanta [11] introduced an anomaly detection strategy in WSNs utilizing ensemble random forest (ERF), with Decision Tree, Naive Bayes, and K-Nearest Neighbor as the ensemble's base learners. The random forest was also built using bootstrap sampling. The authors tested the ERF algorithm on a real-world sensor dataset, namely the activity identification based on multi-sensor data fusion (AReM) dataset.

Dong et al. [12] proposed an intrusion detection model based on the information gain ratio and the bagging algorithm for detecting DoS attacks in cluster-based WSNs. To eliminate unnecessary features, the authors used the information gain ratio. The Bagging algorithm was used to build an ensemble algorithm that trained a series of C4.5 decision trees in order to improve them. To test the model's accuracy, the proposed model was implemented using both the NSL-KDD and the WSN-DS datasets separately.

Otoum et al. [13] proposed a novel methodology for detecting attacks in WSNs that uses an ensemble classifier with Random Forest (RF), Density-Based Spatial Clustering of Applications with Noise (DBSCAN), and Restricted Boltzmann Machine (RBM) as basis classifiers. As a combination technique, Bayesian Combination Classification (BCC) has been used. For performance comparison, Independent BCC (IBCC) and Dependent BCC (DBCC) have been examined. The performance comparisons state that the ensemble technique of DBCC-based IDS shows a promising result over the individual methods in attack detection.

Malmir and Rezvani [14] proposed a novel ensemble approach for anomaly detection in WSNs using Time-overlapped Sliding Windows. Evaluation results confirmed that the proposed method has a strong ability for attack classification and effectively improves the security system in terms of convenient metrics in the area of anomaly detection systems.

Kumari and Mehta [15] developed an ensemble-based model for intrusion detection by combining these two machine learning techniques, J48 DT and SVM. The KDD99 intrusion detection dataset was optimized using particle swarm optimization to identify the nine most relevant and critical attributes, and WEKA was utilized to implement classification. The suggested model yielded results with a higher accuracy of 99.1 % and a lower FAR of 0.9 %.

Fitni and Ramli [16] proposed an ensemble-based AIDS model using DT, LR, and gradient boosting as inputs to an ensemble learning stacking classifier. The Chi-squared correlation approach was used to determine 23 relevant characteristics from the Communications Security Establishment and Canadian Institute for Cybersecurity 2018 (CSE-CIC-IDS2018) dataset. With 98.8 % percent accuracy and a 97.1 % percent detection rate score, the proposed model outperforms seven individual classifiers.

However, none of the aforementioned works take into account continuous streaming data, and rarely has work been done for anomaly detection for WSNs based on online ensemble learning in real-time. There has been little research into anomaly detection in streaming data for embedded systems.

The work in [17] by authors Bosman et al. presents a new lightweight architecture focused on ensembles of incremental learners for online anomaly detection in IoT applications, including WSNs. Also in environments with little a priori knowledge, their decentralized methodology outperformed each individual centralized offline learner alternative in detecting anomalies, determining that ensemble schemes are realistic to adopt.

Ding et al. [18] proposed a distributed online ensemble anomaly detector method in resource-constrained WSNs. Ensemble pruning based on biogeographical-based optimization (BBO) was employed to reduce the high resource demand and produce an optimized detector that performs at least as well as the original ones. The experiments operated on a real WSN dataset and demonstrated the effectiveness of the proposed method.

Alrashdi et al. [19] proposed a framework for identifying attacks in the fog node by employing the Online Sequential Extreme Learning Machine (OS-ELM) and majority voting to discover anomalies. The authors utilized the NSL-KDD to analyze and test their framework.

Ifzarne et al. [20] proposed an online learning classifier utilizing the information gain ratio to choose the relevant features of the sensor data with an online Passive aggressive algorithm in order to identify different types of DoS attacks. The WSN-DS dataset was utilized by the authors for the experiment.

Martindale et al. [21] proposed an approach for detecting intrusions in IoTs by exploring the performance and run-time trade-offs of a set of several online individual algorithms as well as a few homogeneous and heterogeneous ensemble approaches. The massive online analysis (MOA) framework was used for implementing their approach. The 11 algorithms were run against three different KDDCup99 subsets. This study demonstrated that the ensembles outperformed the individual base learners, but at a higher cost in terms of run time, and the heterogeneous ensemble, which consisted of an ARF combined with HAT, outperformed the other online ensembles.

Although there are numerous studies exploring the use of online ensemble approaches and applying machine learning methods to streams of data, the majority of them ignore resource constraints and are targeted for Internet-of-Things (IoT) devices rather than WSNs.

3 RESEARCH METHODOLOGY

3.1 WSN Network Topology Based on LEACH Routing Protocol

Many researchers used IDS to perform their work for WSN. Their work varies depending on the WSN topology and the protection method used, according to [2]. WSN topologies are primarily divided into two types: flat-based and cluster-based. The LEACH (Low Energy Adaptive Clustering Hierarchy) protocol is one of the main proactive sensor network protocols and a widely used clustering technique in WSN. LEACH was proposed by Wendi B. Heinzelman of MIT [22]. It is a self-organizing, hierarchical routing protocol based on adaptive clustering that is used in WSN to reduce the energy consumption of network elements in order to prolong their lifetime [23, 24, 25]. The LEACH consists of three parts: Sensor nodes (SN), cluster head (CH) nodes, and base stations (BS). The LEACH protocol's key concept is to group nodes into clusters in order to spread energy among all nodes in the network. The CH nodes gather and process the SN data in the cluster and

transmit it to the base station. The cluster head nodes can monitor the behavior of the network traffic passing by in real time, and the intrusion detection model can be deployed as a purely centralized one where the IDS is installed at the base station. BS detects intrusions by analyzing the monitored network activity data.

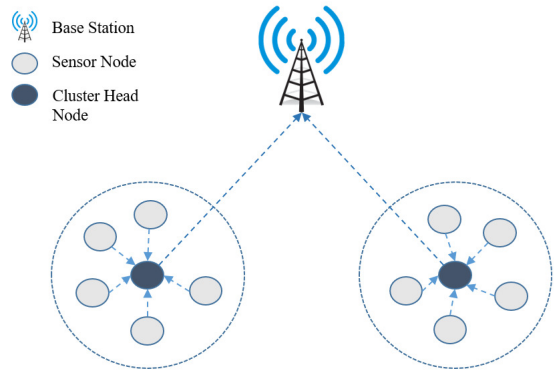


Figure 1. WSN network topology

3.2 Classification Algorithms

There are numerous models and algorithms for machine learning, according to the "no free lunch" (NFL) theorem stated by Wolpert and McReady [26] claim that in ML "there is no single learning algorithm that universally performs best across all domains" [27]. The NFL theorem emphasizes the effectiveness of experimenting with various machine classifiers while tackling classification problems. Testing different classifiers is the most precise technique to solve domain-specific problems; in our situation, the problem is intrusion detection. We will focus on two types of classification algorithms: single classifiers and ensembles.

3.2.1 Single Classifier

Support Vector Machine (SVM): Is a supervised learning algorithm that can be used for both linear and nonlinear problems. In SVMs, the idea is to find a max-margin separation hyperplane in the n-dimensional feature space. Because the separation of hyperplanes is determined by a small number of support vectors, SVMs can yield satisfying results with small-scale training sets. On the other hand, SVMs are sensitive to noise around the hyperplane.

k-Nearest Neighbors (KNN): Is a supervised learning algorithm that is very simplistic and can be used to fill in missing values and resample datasets. The core idea is to predict the class of a data sample using "feature similarity". In the KNN algorithm, the calculation of the distance from the neighbors is used

to classify a sample based on its neighbors. The performance of KNN models is heavily influenced by the parameter k. As long as the value of K is very small, the more complex the model, the higher the risk of overfitting. On the other hand, the larger k, the simpler the model and the lower the fitting ability [28, 29]. The value of K in our study is equal to 10.

Naive Bayes (NB): Is a well-known classification technique that is based on conditional probability and the concept of attribute independence and uses Bayes' theorem to forecast the likelihood or probability of an event occurring based on previous observations of related events [30]. This algorithm is robust to noise and able to learn incrementally; on the other hand, the NB method performs poorly on attribute-related data.

Passive Aggressive (PA): Is a family of online learning algorithms generally used for large-scale learning that can be adopted for both classification and regression challenges. It is similar to the SVM classifier. The PA classifier attempts to find hyperplanes that can be used to split the instances into halves [31].

Perceptron (P): Is a supervised learning algorithm for classification, and it is probably the most basic type of neural network model. The algorithm is made up of a single node or neuron that processes a row of data and predicts a class label.

Hoeffding Tree (HT): An extremely fast decision tree technique for streaming data has been proposed by Domingos and Hulten [32] in which we wait for new instances to arrive rather than reusing instances to compute the best splitting attributes. The HT's most intriguing characteristic is that it constructs a tree that provably converges to the tree constructed by a batch learner with suitably large data.

Hoeffding Adaptive Tree (HAT): Adaptive tree monitors the performance of branches on the tree using the adaptive windowing (ADWIN) algorithm [33] to recent data and replaces them with new branches when their accuracy declines if the new branches are more accurate.

3.2.2 Ensemble Classifier

When compared to a single model, ensemble learning enhances machine learning results by combining several models to improve predictive performance. There are two possibilities for an ensemble of classifiers:

Homogeneous Ensembles: An ensemble of the same type of classifiers. A well-known example of this is random forests. A random forest is a homogeneous ensemble that is a collection of many individual decision trees.

1. Adaptive Random Forest (ARF): A streaming classifier devoted to evolving data streams, originally proposed in [34]. ARF uses a similar approach to the classical Random Forest algorithm [35]. In our approach, the ARF was

performed with the default settings, which consisted of ten HTs algorithms as base learners and included a drift detection operator. ARF is made up of 20 HT algorithms and is tagged "ARF (20)".

2. Online Bagging: Authors in [36] proposed the online bagging algorithm, which employs a weighting approach to approximate the initial random sampling with replacement of instances, giving each arriving example a weight according to Poisson (1).

Heterogeneous Ensembles: An ensemble of different classifiers, which, for example, can contain SVM classifiers, neural network classifiers, and decision trees all at the same time.

1. Weighted Majority (WM): A simple but widely studied algorithm proposed by [37] that makes predictions based on a series of expert advices and learns to adjust its weights over time.

3.3 The Proposed Online Ensemble Intrusion Detector Model for WSNs

The experiments in this work are designed to provide guidance on the appropriate ensemble technique for complying with the requirements of an ideal IDS for WSNs, and this is accomplished by comparing the output of homogeneous ensembles composed of the same algorithm to build all the classifiers with that of heterogeneous ensembles composed of different algorithms. In our approach, we propose online ensemble classification methods that attain a higher detection rate with the aim of distinguishing malicious attacks while taking into account the resource constraints of WSNs.

Designing an ensemble often lies in two main challenges, which are the choice of available base classifiers and the combiner methods:

- The detection of intrusions in our study consists of using online supervised learning, and the performance is measured through the use of the prequential (test-then-train approach) evaluation method, which was developed specifically for stream applications, where each sample is tested and then trained on in sequence by constructing a prediction with our current model for each incoming observation in the stream and scoring that prediction based on how well it matches the actual observation. The online model is then updated with the observation, and we proceed to the next instance. This evaluation technique is discussed in greater detail in the following section.
- We have used WSN-DS, a well-known dataset built exclusively for WSNs, to improve DoS detection and classification, which has been used in previous research such as [9, 38, 12], and we investigated the application of seven stand-alone algorithms, i.e., SVM, HAT, KNN, NB, PA, P, and HT, to detect intrusions in WSNs.

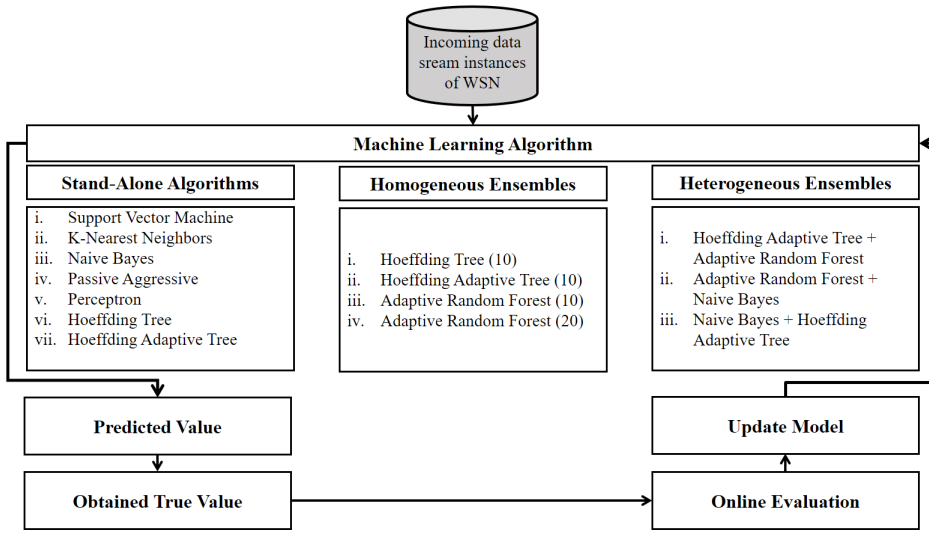


Figure 2. The structure of the proposed approach

- Second, we inspected different homogeneous ensemble approaches, such as HAT and ARF, using an online bagging algorithm. The final online bagging model predicts the simple majority vote of the basis classifier predictions.
- The performance of the ensemble is determined by two properties: The individual success of the ensemble's base classifiers and the independence of the base classifiers results from one another. We compared numerous online classifiers to determine the best appropriate classifiers for online ensemble learning, and based on the results, the algorithms (HAT, ARF, and NB) were chosen as base learners for the heterogeneous ensembles since they provided the highest predictive accuracy on streaming data. We employed a majority vote ensemble to combine the results of two different stable learners in order to achieve both speed and precision at the same time. Three heterogeneous ensembles are proposed: HAT + ARF, ARF + NB, and NB + HAT.

4 EXPERIMENTAL ENVIRONMENT

4.1 Experimental WSN-DS Dataset

In this work, we utilized the simulated wireless sensor network-detection system (WSN-DS) dataset developed by [38], and the Network Simulator NS-2 was used to simulate the wireless sensor network environment [39]. Based on the LEACH routing protocol the required data from the network with different attack scenarios are gathered. The dataset consists of 23 features identifying the state of each sensor.

The LEACH protocol was selected because it is one of the most well-known energy-efficient clustering algorithms for WSNs. The number of samples within the WSN-DS dataset is 374 661. Four types of Denial of Service (DoS) attacks are implemented in the constructed dataset: Blackhole, Grayhole, Flooding, and Scheduling attacks, in addition to the normal behavior (no-attack) records. Using the label encoding technique, all of the categorical values of the label feature in the sample data are converted to numeric values to eliminate their impact on the algorithms.

Simulation parameters are summarized in Table 1. The distribution of the data is illustrated in Table 2. Though only 19 dimensional feature data involving the class label were in the dataset file, as shown in Table 3. The following are the technical characteristics of the computer used throughout the implementation phase:

- Central Processing Unit: Intel (R) Core (TM) i7-4610M CPU @ 3.00 GHz;
- Random Access Memory: 8 GB;
- Operating System: Windows 7 Pro 64-bit.

Parameter	**Value**
Number of nodes	100 nodes
Number of clusters	5
Network area	100 m × 100 m
Base station location	(50 175)
Size of packet header	25 bytes
Size of data packet	500 bytes
Maximum transmission range	200 m
MAC protocol	CSMA/TDMA
Routing protocol	LEACH
Simulation time	3 600 s

Table 1. WSN Simulation parameters

Types of Attack	**Quantity**	**Proportion (%)**
Normal	340 066	90.77
Grayhole	14 596	3.90
Blackhole	10 049	2.68
Scheduling	6 638	1.77
Flooding	3 312	0.88

Table 2. Distribution of WSN-DS dataset

4.2 The Scikit-Multiflow Framework

The experimental setup of this research aims to provide reproducibility, allowing different researchers to achieve with a high degree of agreement the same results obtained in this experiment by using the same experimental framework. However, the

Feature Number	Symbol	Feature Name	Description
1	id	Node Id	A unique ID number of the sensor node
2	Time	Time	The run-time of the node in the simulation
3	Is_CH	Is CH	Describes if the node is a CH or not
4	Who_CH	Who CH	Cluster head ID
5	Dist_To_CH	Distance to CH	Distance between node and CH
6	ADV_S	ADV CH sends	Number of the advertise CH's broadcast messages sent to nodes
7	ADV_R	ADV CH receives	Number of advertise messages received by the nodes from CH
8	JOIN_S	Join request send	Number of join request messages sent by the nodes to the CH
9	JOIN_R	Join request receive	Number of join request messages received by CH from nodes
10	SCH_S	ADV SCH sends	Messages of TDMA schedule broadcast sent to the nodes
11	SCH_R	ADV SCH receives	Number of scheduled messages received by the CH
12	Rank	Rank	Node order in TDMA scheduling
13	DATA_S	Data sent	Number of data packets sent from the node to its CH
14	DATA_R	Data received	Number of data packets received by the node from the CH
15	Data_Sent_To_BS	Data sent to BS	Number of data packets that are sent from node to the BS
16	dist_CH_To_BS	Distance CH to BS	Distance between CH and BS
17	send_code	Send code	The sending code of the cluster
18	Consumed_Energy	Energy consumption	Energy consumed
19	Attack type	Attack type	Type of attacks or normal traffic

Table 3. Features of the WSN-DS dataset

current experimental machine learning tooling is mainly divided between Java-based and Python-based implementations. Some researchers implement their experiments using tools built around Weka [40] or the data stream mining framework MOA [41]. Others prefer solutions from the scikit-learn environment or the multioutput streaming platform scikit-multiflow [42].

Scikit-multiflow is a Python framework for learning from data streams and multi-output/multi-label learning. Scikit-multiflow is based on well-known open-source frameworks like scikit-learn, MOA, and MEKA. It includes techniques for classification, regression, concept drift detection, and anomaly detection. It also has

a collection of data stream generators and evaluators. scikit-multiflow is intended to work with Python's numerical and scientific libraries, NumPy and SciPy, as well as Jupyter Notebooks.

4.3 Evaluation Technique

Considering the unbounded real-world streams of non-stationary data, classic techniques for evaluating the model on batch data such as train-test split and cross-validation do not apply to models trained on streamed data [43]. To succeed in dealing with this problem and obtaining accurate measurements, over time, we used the widely known prequential evaluation method in our experiment.

The Prequential evaluation method or the interleaved test-then-train method. Each individual sample is used to test the model, which means to make predictions, and then the same sample is used to train the model, and from this the accuracy can be incrementally updated. When the evaluation is intentionally performed in this order, the model is always tested on instances that it has not yet seen. This approach has the favorable condition that no holdout set is needed for testing, making maximum use of the available data.

4.4 Evaluation Metrics

The adequacy of the wireless sensor network intrusion detection algorithms was measured using the following measures: Accuracy (Acc), precision (P), recall (R), F1-Score (F), as well as the total running time (Training Time + Testing Time) in seconds (s) of the classification algorithm are collected and compared. Table 4 shows the definitions of TP, FP, TN and FN.

An abnormal flow is treated as Positive (P) and a normal flow is treated as Negative (N).

True Positive: The model correctly predicts the positive class. The model correctly predicts that an instance is an attack by the classifier.

True Negative: The model correctly predicts the negative class. The model correctly predicts that an instance is normal.

False Positive: The model incorrectly predicts the positive class. The data samples (attack) were incorrectly predicted as normal.

False Negative: The model incorrectly predicts the negative class. The normal data samples were incorrectly predicted as an attack.

These performance evaluation metrics are calculated as follows:

Accuracy (Acc): Represents the percentage of instances correctly classified. This is the number of correct predictions divided by the total number of predictions. Accuracy alone is not sufficient as a measure of performance, especially for

	Positive real class (Abnormal)	**Negative real class (Normal)**
Positive predicted class (Abnormal)	True Positive (TP)	False Negative (FN)
Negative predicted class (Normal)	False Positive (FP)	True Negative (TN)

Table 4. Definition of TP, FP, TN and FN

datasets with unbalanced classes. It can be written as:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{1}$$

Precision (P): Or Positive Predictive value (PPV) represents how good the model is at assigning positive events to the positive class. That is, how accurate the attack prediction is, and it is defined as follows:

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

Recall (R): It is also called the "True Positive Rate (TPR)", "sensitivity" and "detection rate", and it measures how good the model is at detecting the positive class. So, given that attacks are the positive class, it represents the percentage of actual attacks that were correctly identified. Equation (3) represents the formula for calculating recall:

$$Recall = DetectionRate = \frac{TP}{TP + FN} \tag{3}$$

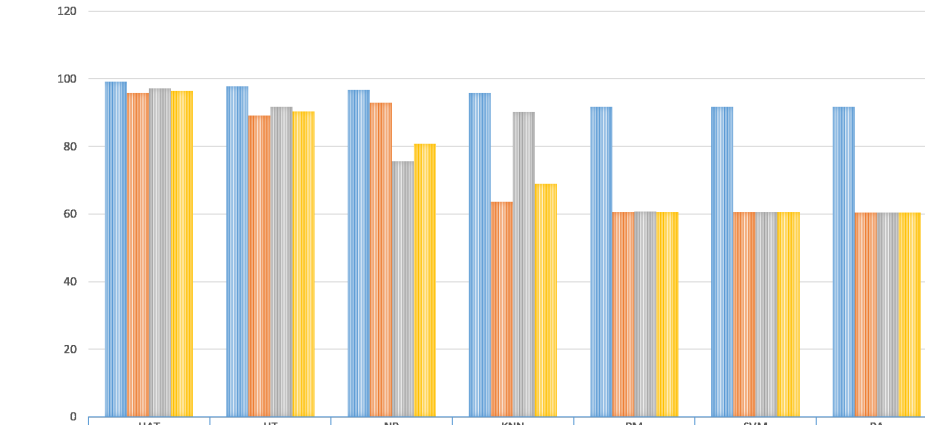
F1-Score (F): Or F1-Measure represents the harmonic mean of Precision and Recall. Compared to accuracy, the f1-score is the best metric to check the effectiveness of an intrusion detection algorithm when the IDS model uses an unbalanced dataset and we search for a balance between precision and recall.

$$F1\text{-}Score = \frac{2 \times P \times R}{P + R} \tag{4}$$

5 EXPERIMENTAL RESULTS

5.1 Individual Algorithms

This subsection presents and discusses the performance evaluation results of each algorithm when running individually. As can be seen from Figure 3, the R of the HAT method reaches 95.86 % which is the highest detection rate compared with other online methods. The Acc of this method is 99.14 %, and the P and F are respectively 97.17 % and 96.48 %. In second place, Naive Bayes had a detection rate



	HAT	HT	NB	KNN	PM	SVM	PA
Accuracy(%)	99,14	97,72	96,64	95,85	91,76	91,75	91,74
Recall(%)	95,86	89,13	92,98	63,62	60,56	60,64	60,46
Precision(%)	97,17	91,75	75,6	90,25	60,66	60,61	60,46
F1-score(%)	96,48	90,32	80,81	68,92	60,61	60,54	60,46

Figure 3. Comparison of performance of different individual models results

Models	**Run-time (s)**
KNN	1 089.67
PA	881.58
PM	785.87
SVM	781.18
HAT	218.08
NB	141.95
HT	100.28

Table 5. Running time (Training Time + Testing Time) in seconds (s)

of 92.98 % followed by the HT method with a R of 89.13 %. In summary, the HAT method performs better than the other online intrusion detection methods.

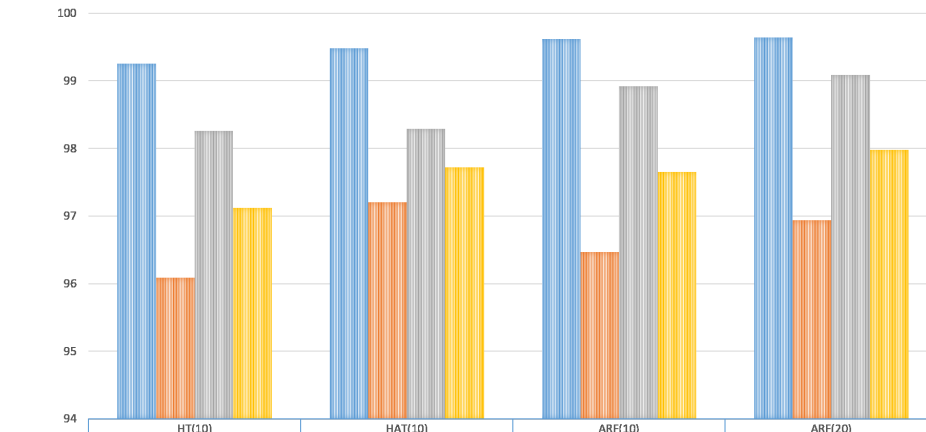
Comparing the model total running time, we can see that the HT had the fastest run-times, followed by the model NB, and in third place, the HAT method has a lower time than that of KNN, PM, SVM, and PA. On the other hand, the KNN model exhibits a considerably longer running time of approximately 18 minutes compared with other tested methods. The necessity to continuously calculate the distance between the predicted target and every other sample still in memory by the KNN algorithm might give an explanation for the long running time. In some critical streaming applications, such as predicting network security intrusions, a fast but less accurate model is often preferred over a slow but more accurate model.

5.2 Ensemble Algorithms

5.2.1 Homogeneous Ensembles

Figure 4 displays diverse homogeneous ensemble approaches, where each ensemble is made up of multiple instances of the same base learner. All of the ensemble approaches that have been tested have produced excellent results, particularly HAT (10), which has a detection rate (R) of 97.2 %. Immediately afterwards, ARF (20) with a R that attains 96.94 %. When comparing Figure 3 with Figure 4 in terms of results concerning HT, HT (10), HAT and HAT (10), as indicated by the homogeneous ensemble results, combining numerous online learnables for prediction enhances the detection of attacks greatly when compared to single predictors. Increasing the number of trees substantially enhances performance, as is the case when comparing the results of AFR (10 and 20).

We can observe from Table 6 that HT (10) is faster than the rest of the online homogeneous ensembles and ARF (20) has a higher classification time. Our results highlight that the running time of the ensemble algorithms is significantly longer than the individual methods when comparing Table 5 with Table 6. The number of estimators does have an impact on the model's running time.



	HT(10)	HAT(10)	ARF(10)	ARF(20)
Accuracy(%)	99,25	99,48	99,62	99,64
Recall(%)	96,09	97,2	96,47	96,94
Precision(%)	98,26	98,29	98,92	99,09
F1-score(%)	97,12	97,72	97,65	97,98

Figure 4. Homogeneous ensembles results

5.2.2 Heterogeneous Ensembles

Restating that the ARF was run with the default settings, which included 10 HTs algorithms. As can be seen from Figure 5 which presents the predictive performance

Models	Run-time (s)
HT (10)	971.95
HAT (10)	2 400.01
ARF (10)	2 049.89
ARF (20)	4 214.41

Table 6. Running time metric in seconds (s)

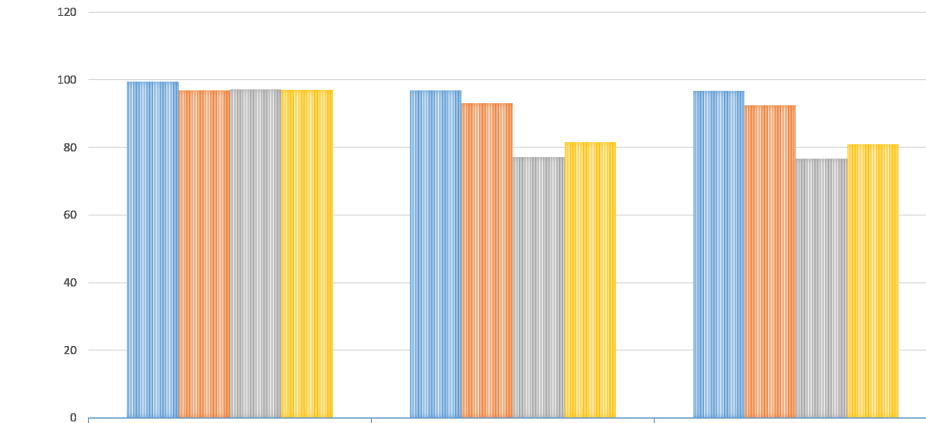


Figure 5. Heterogeneous ensembles results

of the heterogeneous ensembles, the online evaluation results show that the detection rate of the method (ARF + HAT) in this paper reaches 96.84 %, which is higher than that of (ARF + NB) and (HAT + NB). The Acc, P, and F are respectively 99.42 %, 97.23 % and 96.96 %. According to the performance comparisons of ARF with a R of 96.47 % and HAT with a R of 95.86 %, the heterogeneous ensemble of (ARF + HAT) methodology outperforms the seven stand-alone algorithms and the homogeneous ensemble of ARF (10) in overcoming the misclassification of attacks. The reason is that the accuracy of individual models and the diversity among individual models are all aspects that contribute to the heterogeneous ensemble's success.

Models	Run-time (s)
ARF + HAT	2 345
ARF + NB	2 192
HAT + NB	403

Table 7. Running time metric in seconds (s)

We can see from Table 7 that the heterogeneous ensemble of the (HAT + NB) technique has a relatively short running time as it contains two base learners. On the other hand, the (ARF + HAT) and (ARF + NB) classification times are close to each other, with the (ARF + HAT) method being more time-consuming and having the correspondingly highest detection rate.

5.2.3 Concept Drift Results

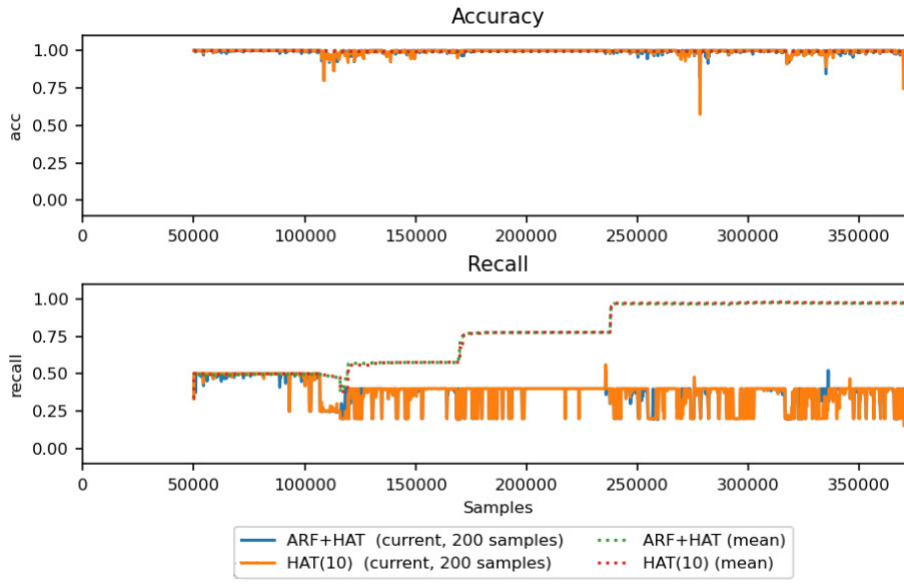


Figure 6. Concept drift results

The existence of concept drift in streaming data is a significant element that contributes to a decrease in predictive accuracy over time. Figure 6 depicts a plot of the predictive accuracy offered over time for the homogeneous ensemble HAT (10) as well as the heterogeneous ensemble ARF+HAT presented by (orange) and (blue), respectively, both of which performed favorably in terms of accuracy and classification of attacks. In terms of concept drift, however, there is no evident performance advantage between these two. While the HAT (10) ensemble's classification performance suffers from significant dips in our results, there are several small areas where the HAT (10) ensemble tackles concept drifts more effectively than the ARF + HAT. Even though the model performance of HAT (10) was similar to the combined ARF + HAT, the results demonstrated that the performance of the model alone does not necessarily indicate how an algorithm responds to concept drift.

Based on the experimental findings, the following conclusions can be drawn:

- The results showed that nearly all ensemble techniques, both homogeneous and heterogeneous, significantly outperformed single classification models in terms of Acc, R, P, and F1 performance measures, although at the expense of increasing run-time.
- Thus, based on our experiments, depending on specific application requirements and resource constraints, one can note by comparing the models' performances of the seven ensembles that the best model can be either the heterogeneous ensemble (ARF+HAT) or the homogeneous ensemble HAT (10), since both have delivered the highest predictive performance and overcome the misclassification of attacks in WSNs.
- The online ensemble algorithm demonstrates great abilities to continuously process traffic data on a large scale because, while the classifier learns and attempts to train the model better, the detection rate of attacks continues to improve over time/iterations as made evident by the improvement in recall, and it is also applicable to any application, making it advantageous compared to existing models that are specific to their applications.

6 CONCLUSION

Wireless sensor networks are often referred to as an emerging technology that will impact our daily lives. These electromechanical components, which are very small and communicate via a ubiquitous wireless network, widely open the horizons of applications built up to now. Being exposed to numerous risks, the main challenge of the evolution of intrusion detection systems in WSNs is to identify the attacks with great accuracy and respond to the constraints and challenges required to extend the life of the entire network. Given that much more attention is paid to the detection techniques used, this goal could be achieved in a variety of ways.

Our paper presents a novel perspective on the malicious security attacks in WSNs by involving a fast intrusion detection scheme based on ensemble learning that satisfies the dynamic and continuous streaming of data. Our experiments with the WSN-DS attack database show that the ensemble approach performed better than any online classifier as an individual learner, despite having a generally longer run-time in distinguishing attacks from benign samples; thus, we propose heterogeneous ensemble (ARF + HAT) and homogeneous ensemble HAT (10), as both achieve higher detection rates in the aim of distinguishing malicious attacks while taking into account the resource constraints of WSNs when compared to other intrusion detection methods, and its prediction performance improves over time as new data points are integrated. In general, our proposed model is effective for real-time WSN intrusion detection.

In future work, we will explore different methods, such as preprocessing with data reduction and parameter tuning, that can improve the efficiency of the classifier.

The performance can be further improved using deep learning techniques to enhance WSN's attack detection performance.

REFERENCES

[1] Marriwala, N.—Rathee, P.: An Approach to Increase the Wireless Sensor Network Lifetime. 2012 World Congress on Information and Communication Technologies, IEEE, 2012, pp. 495–499, doi: 10.1109/WICT.2012.6409128.

[2] Osanaiye, O. A.—Alfa, A. S.—Hancke, G. P.: Denial of Service Defence for Resource Availability in Wireless Sensor Networks. IEEE Access, Vol. 6, 2018, pp. 6975–7004, doi: 10.1109/ACCESS.2018.2793841.

[3] Salmon, H. M.—de Farias, C. M.—Loureiro, P.—Pirmez, L.—Rossetto, S.—de A. Rodrigues, P. H.—Pirmez, R.—Delicato, F. C.—da Costa Carmo, L. F. R.: Intrusion Detection System for Wireless Sensor Networks Using Danger Theory Immune-Inspired Techniques. International Journal of Wireless Information Networks, Vol. 20, 2013, No. 1, pp. 39–66, doi: 10.1007/s10776-012-0179-z.

[4] Abduvaliyev, A.—Pathan, A. S. K.—Zhou, J.—Roman, R.—Wong, W. C.: On the Vital Areas of Intrusion Detection Systems in Wireless Sensor Networks. IEEE Communications Surveys and Tutorials, Vol. 15, 2013, No. 3, pp. 1223–1237, doi: 10.1109/SURV.2012.121912.00006.

[5] Mitrokotsa, A.—Karygiannis, A.: Intrusion Detection Techniques in Sensor Networks. In: Lopez, J., Zhou, J. (Eds.): Wireless Sensor Network Security. IOS Press, 2008, pp. 251–272.

[6] Opitz, D.—Maclin, R.: Popular Ensemble Methods: An Empirical Study. Journal of Artificial Intelligence Research, Vol. 11, 1999, pp. 169–198, doi: 10.1613/jair.614.

[7] Sen, J.—Mehtab, S.: Machine Learning Applications in Misuse and Anomaly Detection. In: Kalloniatis, C., Travieso-Gonzalez, C. (Eds.): Security and Privacy from a Legal, Ethical, and Technical Perspective. IntechOpen, Rijeka, 2020, doi: 10.5772/intechopen.92653.

[8] Zamry, N. M.—Zainal, A.—Rassam, M. A.—Alkhammash, E. H.—Ghaleb, F. A.—Saeed, F.: Lightweight Anomaly Detection Scheme Using Incremental Principal Component Analysis and Support Vector Machine. Sensors, Vol. 21, 2021, No. 23, Art. No. 8017, doi: 10.3390/s21238017.

[9] Alqahtani, M.—Gumaei, A.—Mathkour, H.—Maher Ben Ismail, M.: A Genetic-Based Extreme Gradient Boosting Model for Detecting Intrusions in Wireless Sensor Networks. Sensors, Vol. 19, 2019, No. 20, Art. No. 4383, doi: 10.3390/s19204383.

[10] Le, T. T. H.—Park, T.—Cho, D.—Kim, H.: An Effective Classification for DoS Attacks in Wireless Sensor Networks. 2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN), IEEE, 2018, pp. 689–692, doi: 10.1109/ICUFN.2018.8436999.

[11] Biswas, P.—Samanta, T.: Anomaly Detection Using Ensemble Random Forest in Wireless Sensor Network. International Journal of Information Technology, Vol. 13, 2021, No. 5, pp. 2043–2052, doi: 10.1007/s41870-021-00717-8.

[12] Dong, R. H.—Yan, H. H.—Zhang, Q. Y.: An Intrusion Detection Model for Wireless Sensor Network Based on Information Gain Ratio and Bagging Algorithm. International Journal of Network Security, Vol. 22, 2020, No. 2, pp. 218–230, doi: 10.6633/IJNS.202003_22(2).05.

[13] Otoum, S.—Kantarci, B.—Mouftah, H. T.: A Novel Ensemble Method for Advanced Intrusion Detection in Wireless Sensor Networks. ICC 2020, 2020 IEEE International Conference on Communications (ICC), 2020, pp. 1–6, doi: 10.1109/ICC40277.2020.9149413.

[14] Malmir, Z.—Rezvani, M. H.: A Novel Ensemble Approach for Anomaly Detection in Wireless Sensor Networks Using Time-Overlapped Sliding Windows. Journal of Computer and Robotics, Vol. 12, 2019, No. 1, pp. 1–13.

[15] Kumari, A.—Mehta, A. K.: A Hybrid Intrusion Detection System Based on Decision Tree and Support Vector Machine. 2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA), 2020, pp. 396–400, doi: 10.1109/ICCCA49541.2020.9250753.

[16] Fitni, Q. R. S.—Ramli, K.: Implementation of Ensemble Learning and Feature Selection for Performance Improvements in Anomaly-Based Intrusion Detection Systems. 2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT), 2020, pp. 118–124, doi: 10.1109/IAICT50021.2020.9172014.

[17] Bosman, H. H. W. J.—Iacca, G.—Tejada, A.—Wörtche, H. J.—Liotta, A.: Ensembles of Incremental Learners to Detect Anomalies in Ad Hoc Sensor Networks. Ad Hoc Networks, Vol. 35, 2015, pp. 14–36, doi: 10.1016/j.adhoc.2015.07.013.

[18] Ding, Z.—Wang, H.—Fei, M.—Du, D.: A Novel Distributed Online Anomaly Detection Method in Resource-Constrained Wireless Sensor Networks. International Journal of Distributed Sensor Networks, Vol. 11, 2015, No. 10, Art. No. 146189, doi: 10.1155/2015/146189.

[19] Alrashdi, I.—Alqazzaz, A.—Alharthi, R.—Aloufi, E.—Zohdy, M. A.—Ming, H.: FBAD: Fog-Based Attack Detection for IoT Healthcare in Smart Cities. 2019 IEEE 10th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON), 2019, pp. 0515–0522, doi: 10.1109/UEMCON47517.2019.8992963.

[20] Ifzarne, S.—Tabbaa, H.—Hafidi, I.—Lamghari, N.: Anomaly Detection Using Machine Learning Techniques in Wireless Sensor Networks. Journal of Physics: Conference Series, Vol. 1743, 2021, No. 1, Art. No. 012021, doi: 10.1088/1742-6596/1743/1/012021.

[21] Martindale, N.—Ismail, M.—Talbert, D. A.: Ensemble-Based Online Machine Learning Algorithms for Network Intrusion Detection Systems Using Streaming Data. Information, Vol. 11, 2020, No. 6, Art. No. 315, doi: 10.3390/info11060315.

[22] Heinzelman, W. R.—Chandrakasan, A.—Balakrishnan, H.: Energy-Efficient Communication Protocol for Wireless Microsensor Networks. Proceedings of the 33rd

Annual Hawaii International Conference on System Sciences, IEEE, Vol. 2, 2000, pp. 1–10, doi: 10.1109/HICSS.2000.926982.

[23] Xu, J.—Jin, N.—Lou, X.—Peng, T.—Zhou, Q.—Chen, Y.: Improvement of LEACH Protocol for WSN. 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery, IEEE, 2012, pp. 2174–2177, doi: 10.1109/FSKD.2012.6233907.

[24] Liu, H.—Li, L.—Jin, S.: Cluster Number Variability Problem in LEACH. In: Ma, J., Jin, H., Yang, L. T., Tsai, J. J. P. (Eds.): Ubiquitous Intelligence and Computing. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 4159, 2006, pp. 429–437, doi: 10.1007/11833529_44.

[25] Heinzelman, W. B.—Chandrakasan, A. P.—Balakrishnan, H.: An Application-Specific Protocol Architecture for Wireless Microsensor Networks. IEEE Transactions on Wireless Communications, Vol. 1, 2002, No. 4, pp. 660–670, doi: 10.1109/TWC.2002.804190.

[26] Wolpert, D. H.—Macready, W. G.: No Free Lunch Theorems for Optimization. IEEE Transactions on Evolutionary Computation, Vol. 1, 1997, No. 1, pp. 67–82, doi: 10.1109/4235.585893.

[27] Douglas, P. K.—Harris, S.—Yuille, A.—Cohen, M. S.: Performance Comparison of Machine Learning Algorithms and Number of Independent Components Used in fMRI Decoding of Belief vs. Disbelief. NeuroImage, Vol. 56, 2011, No. 2, pp. 544–553, doi: 10.1016/j.neuroimage.2010.11.002.

[28] Ma, Z.—Kaban, A.: K-Nearest-Neighbours with a Novel Similarity Measure for Intrusion Detection. 2013 13th UK Workshop on Computational Intelligence (UKCI), IEEE, 2013, pp. 266–271, doi: 10.1109/UKCI.2013.6651315.

[29] Cover, T.—Hart, P.: Nearest Neighbor Pattern Classification. IEEE Transactions on Information Theory, Vol. 13, 1967, No. 1, pp. 21–27, doi: 10.1109/TIT.1967.1053964.

[30] Kotsiantis, S. B.: Supervised Machine Learning: A Review of Classification Techniques. Emerging Artificial Intelligence Applications in Computer Engineering, IOS Press, 2007, pp. 3–24.

[31] Crammer, K.—Dekel, O.—Keshet, J.—Shalev-Shwartz, S.—Singer, Y.: Online Passive Aggressive Algorithms. Journal of Machine Learning Research, Vol. 7, 2006, No. 19, pp. 551–585, http://jmlr.org/papers/v7/crammer06a.html.

[32] Domingos, P.—Hulten, G.: Mining High-Speed Data Streams. Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '00), 2000, pp. 71–80, doi: 10.1145/347090.347107.

[33] Bifet, A.—Gavaldà, R.: Learning from Time-Changing Data with Adaptive Windowing. Proceedings of the 2007 SIAM International Conference on Data Mining (SDM), 2007, pp. 443–448, doi: 10.1137/1.9781611972771.42.

[34] Gomes, H. M.—Bifet, A.—Read, J.—Barddal, J. P.—Enembreck, F.—Pfharinger, B.—Holmes, G.—Abdessalem, T.: Adaptive Random Forests for Evolving Data Stream Classification. Machine Learning, Vol. 106, 2017, No. 9, pp. 1469–1495, doi: 10.1007/s10994-017-5642-8.

[35] Breiman, L.: Random Forests. Machine Learning, Vol. 45, 2001, No. 1, pp. 5–32, doi: 10.1023/A:1010933404324.

[36] Oza, N. C.—Russell, S. J.: Online Bagging and Boosting. In: Richardsom, T. S., Jaakkola, T. S. (Eds.): Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research (PMLR), Vol. R3, 2001, pp. 229–236, https://proceedings.mlr.press/r3/oza01a.html.

[37] Littlestone, N.—Warmuth, M. K.: The Weighted Majority Algorithm. Information and Computation, Vol. 108, 1994, No. 2, pp. 212–261, doi: 10.1006/inco.1994.1009.

[38] Almomani, I.—Al-Kasasbeh, B.—Al-Akhras, M.: WSN-DS: A Dataset for Intrusion Detection Systems in Wireless Sensor Networks. Journal of Sensors, Vol. 2016, 2016, Art. No. 4731953, doi: 10.1155/2016/4731953.

[39] Issariyakul, T.—Hossain, E.: Introduction to Network Simulator 2 (NS2). Introduction to Network Simulator Ns2, Springer, Boston, 2009, pp. 21–40, doi: 10.1007/978-1-4614-1406-3_2.

[40] Holmes, G.—Donkin, A.—Witten, I. H.: WEKA: A Machine Learning Workbench. Proceedings of ANZIIS '94 – Australian New Zealand Intelligent Information Systems Conference, IEEE, 1994, pp. 357–361, doi: 10.1109/ANZIIS.1994.396988.

[41] Bifet, A.—Holmes, G.—Pfahringer, B.—Kranen, P.—Kremer, H.—Jansen, T.—Seidl, T.: MOA: Massive Online Analysis, a Framework for Stream Classification and Clustering. Proceedings of the First Workshop on Applications of Pattern Analysis, Proceedings of Machine Learning Research (PMLR), Vol. 11, 2010, pp. 44–50, https://proceedings.mlr.press/v11/bifet10a.html.

[42] Montiel, J.—Read, J.—Bifet, A.—Abdessalem, T.: Scikit-Multiflow: A Multi-Output Streaming Framework. The Journal of Machine Learning Research, Vol. 19, 2018, No. 1, pp. 2915–2914.

[43] Gama, J.—Sebastião, R.—Rodrigues, P. P.: Issues in Evaluation of Stream Learning Algorithms. Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09), 2009, doi: 10.1145/1557019.1557060.

Hiba Tabbaa is a Ph.D. student at the National School of Applied Sciences in Khouribga (Sultan Moulay Slimane University, Morocco). She received her Master's degree in big data and decision making in 2020 from the same school. Her main research interests include WSN, big data, machine learning, cybersecurity, cloud computing.

Samir Ifzarne received his Engineering degree from the Mohamadia School of Engineers (EMI), Rabat, in 2001. In 2021, he received his Ph.D. degree in security of wireless sensor networks from the University Sultan Moulay Slimane at the National School of Applied Sciences (ENSA) Khouribga. His research interests include WSN, compressed sensing, and homomorphic encryption.

Imad Hafidi is currently a Professor at the National School of Applied Science (ENSA), Khouribga. He is the Head of the Department of Mathematics and Computer Engineering, along with being the Director of the Laboratory of Process Engineering, Computer Science and Mathematics (LIPIM) of ENSA Khouribga. His research interests include WSN, machine learning, big data, equilibria and computer vision.