scientific reports

OPEN

Mitigating malicious denial of wallet attack using attribute reduction with deep learning approach for serverless computing on next generation applications

Amal K. Alkhalifa[1], Mohammed Aljebreen[2], Rakan Alanazi[3]✉, Nazir Ahmad[4], Sultan Alahmari[5], Othman Alrusaini[6], Ali Alqazzaz[7] & Hassan Alkhiri[8]

Denial of Wallet (DoW) attacks are one kind of cyberattack whose goal is to develop and expand the financial sources of a group by causing extreme costs in their serverless computing or cloud environments. These threats are chiefly related to serverless structures owing to their features, such as auto-scaling, pay-as-you-go method, cost amplification, and limited control. Serverless computing, Function-as-a-Service (FaaS), is a cloud computing (CC) system that permits developers to construct and run applications without a conventional server substructure. The deep learning (DL) model, a part of the machine learning (ML) technique, has developed as an effectual device in cybersecurity, permitting more effectual recognition of anomalous behaviour and classifying patterns indicative of threats. This study proposes a Mitigating Malicious Denial of Wallet Attack using Attribute Reduction with Deep Learning (MMDoWA-ARDL) approach for serverless computing on next-generation applications. The primary purpose of the MMDoWA-ARDL approach is to propose a novel framework that effectively detects and mitigates malicious attacks in serverless environments using an advanced deep-learning model. Initially, the presented MMDoWA-ARDL model applies data pre-processing using Z-score normalization to transform input data into a valid format. Furthermore, the feature selection process-based cuckoo search optimization (CSO) model efficiently identifies the most impactful attributes related to potential malicious activity. For the DoW attack mitigation process, the bi-directional long short-term memory multi-head self-attention network (BMNet) method is employed. Finally, the hyperparameter tuning is accomplished by implementing the secretary bird optimizer algorithm (SBOA) method to enhance the classification outcomes of the BMNet model. A wide-ranging experimental investigation uses a benchmark dataset to exhibit the superior performance of the proposed MMDoWA-ARDL technique. The comparison study of the MMDoWA-ARDL model portrayed a superior accuracy value of 99.39% over existing techniques.

Keywords Denial of wallet, Serverless computing, Cloud computing, Deep learning, Secretary bird optimization, Cybersecurity

[1]Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Kingdom of Saudi Arabia. [2]Department of Computer Science, Community College, King Saud University, P.O. Box 28095, Riyadh 11437, Kingdom of Saudi Arabia. [3]Department of Information Technology, Faculty of Computing and Information Technology, Northern Border University, Rafha, Kingdom of Saudi Arabia. [4]Department of Computer Science, Applied College at Mahayil, King Khalid University, Abha, Kingdom of Saudi Arabia. [5]King Abdul Aziz City for Science and Technology (KACST), Cybersecurity Institute, Riyadh, Kingdom of Saudi Arabia. [6]Department of Engineering and Applied Sciences, Applied College, Umm Al-Qura University, Makkah, Saudi Arabia. [7]Department of Computer Science and Artificial Intelligence, College of Computing and Information Technology, University of Bisha, Bisha 67714, Saudi Arabia. [8]Department of Computer Science, Faculty of Computing and Information Technology, Al-Baha University, Al Bahah, Kingdom of Saudi Arabia. ✉email: rakan.nalenezi@nbu.edu.sa

Serverless computing is an application employment structure that gives pay-as-you-go event-based performance[1]. Serverless function platforms offer the framework for employing code to execute through the cloud and describe the event processing logic by using these methods: trigger, action, and event. Serverless computing summaries back-end management from consumers, permitting only minimum admittance for a few fundamental parameters like function runtime timeout and function memory allocation[2]. Functions implemented on platforms like conventional Infrastructure as a Service (IaaS) virtual machine (VM) assistances; nevertheless, the requirement of this VM is accomplished by the platform with function invocation and not by the developer[3]. A container is made for every function invocation and might be recycled for recurrent invocations[4]. In diverse IaaS, you do not pay for the uptime and source dissipated by the function container or VM execution for the runtime of every function, thus the term FaaS. This method permits serverless applications to measure extensively as the application is presented on the serverless platform of the cloud and admittance to its source[5]. One of these safety attacks is the DoW threat, a different Denial-of-Service (DoS) attack performed on serverless platforms[6]. There are two kinds of DoW attacks, namely internal and external. The external attackers will drain money from the victims by frequently invoking the APIs that the victims accidentally reveal, whereas the internal attackers will cause source contention on the function examples like VM or containers of the victims, reducing their programs[7].

By postponing the performance of the target function, an internal DoW threat has dual direct impacts on the target. External and internal DoW describes the latest investigation on DoW. This study recommends that external DoW might be identified and diminished by conventional DoS mitigation models. Nevertheless, diverse DoS and DoW can utilize alternative threat patterns that do not trigger DoS recognition parameters and cannot be volumetric threats, which creates enormous traffic volumes that search, causing damage in a short period[8]. Alternatively, these threats perform for long periods, not to raise suspicion, as with conventional DoS flooding threats[9]. For internal threat recognition, the systems should analyze and monitor the lower-level source utilization of particular bare-metal machines. DL and ML are progressively utilized in cybersecurity aid to prevent and detect cyber threats. With the development of big data, it became gradually complicated for humans to identify and analyze possible attacks[10]. ML models are skilled in vast datasets of known cyber threats and patterns of suspicious action, and they are utilized to identify and respond to novel attacks in the real world. The swift growth of cloud-native and event-driven applications has improved the reliance on serverless architectures, giving scalability and cost-efficiency. However, this shift also exposes systems to new attack surfaces, specifically DoW attacks that utilize resource-triggered billing mechanisms. As serverless platforms abstract away infrastructure management, attackers can misuse function calls, resulting in unexpected costs. This raises the requirement for intelligent, lightweight defence mechanisms. This motivates the development of a DL-based approach with optimized feature selection to detect and reduce such threats effectually in real-time.

This study proposes a Mitigating Malicious Denial of Wallet Attack using Attribute Reduction with Deep Learning (MMDoWA-ARDL) approach for serverless computing on next-generation applications. The primary purpose of the MMDoWA-ARDL approach is to propose a novel framework that effectively detects and mitigates malicious attacks in serverless environments using an advanced deep-learning model. Initially, the presented MMDoWA-ARDL model applies data pre-processing using Z-score normalization to transform input data into a valid format. Furthermore, the feature selection process-based cuckoo search optimization (CSO) model efficiently identifies the most impactful attributes related to potential malicious activity. For the DoW attack mitigation process, the bi-directional long short-term memory multi-head self-attention network (BMNet) method is employed. Finally, the hyperparameter tuning is accomplished by implementing the secretary bird optimizer algorithm (SBOA) method to enhance the classification outcomes of the BMNet model. A wide-ranging experimental investigation uses a benchmark dataset to exhibit the superior performance of the proposed MMDoWA-ARDL technique. The key contribution of the MMDoWA-ARDL technique is listed below.

- The MMDoWA-ARDL model utilizes Z-score normalization to standardize the input features by centring them around the mean and scaling based on standard deviation (SD). This pre-processing step enhances the model's learning efficiency and stability. It also ensures that all features contribute equally during training, improving overall performance.
- The MMDoWA-ARDL approach employs the CSO method to detect and retain the most informative features from the dataset. It removes irrelevant or redundant data by mitigating dimensionality, resulting in a more efficient model. This improves classification accuracy and mitigates computational overhead.
- The MMDoWA-ARDL methodology uses the BMNet-based architecture, which combines Bi-LSTM with multi-head self-attention to capture both temporal dependencies and contextual associations in the data. This approach enables more accurate and adaptive detection of DoW attacks. Using these advanced techniques, the model improves robustness in handling complex attack scenarios.
- The MMDoWA-ARDL method implements the SBOA technique to fine-tune the model's hyperparameters, optimizing its learning process. This approach systematically searches for the optimum configuration, enhancing overall model efficiency and accuracy. SBOA ensures the model operates at its peak performance.
- by improving parameter settings.
- Integrating CSO, BMNet, and SBOA into a unified framework presents a novel approach to attack mitigation by integrating advanced optimization, DL, and attention mechanisms. This novel combination improves feature selection, model robustness, and hyperparameter tuning, addressing complex security challenges. The novelty is in the seamless synergy between these techniques, which has not been explored in the existing literature on such attack scenarios.

Review of literature

Lu et al.[11] projected an intellectually improving localized GNSS monitoring method that incorporates artificial intelligence (AI) and blockchain (BC) technology. Initially, this study utilizes BCT to ensure GNSS monitoring reliability and tamper resistance of the data. It employs a distributed ledger framework for recognizing the decentralization of data transmission and storage, thus improving the anti-threat reliability and ability of the method. Then, the long short-term memory (LSTM) method is used for forecasting and examining the vast number of monitoring data in the real world, allowing the intellectual recognition of GNSS signal deviations and anomalies and presenting real-world timely warnings to enhance the monitoring impacts. In[12], a BC-based Operational Threat Intelligence architecture, OTI-IoT, is projected. During Level 1, this consortium BC system validators utilize the IPS component from smart contracts for access control and attack prevention. Ullah et al.[13] improve the management and access control of electronic health records (EHR) by utilizing purpose-based access control (PBAC), BC, Smart Contracts, and interplanetary file systems (IPFS). de Lima et al.[14] detect economic denial of sustainability (EDoS) attacks by integrating the synthetic minority oversampling technique (SMOTE), edited nearest neighbour rule (ENN), Random Forest (RF), and XGBoost (XGB) classifiers. The classifiers are optimized utilizing nature-inspired metaheuristics, specifically the bat algorithm (BA), and hyperparameter tuning techniques like random and Bayesian search. Manh et al.[15] developed a novel packing model using a single-instruction-multiple-data (SIMD) method, allowing effective training on Homomorphic Encryption (HE)-encrypted data. Developing this deep neural network (DNN) training model enhanced for encoded data. A privacy-preserving distributed learning method depending on the FedAvg model was projected that parallelizes the training through several workers, substantially enhancing the computation period. Sangore and Patil[16] improve threat detection in blockchain-based Bitcoin systems by integrating ensemble learning, water cycle algorithm (WCA), and dynamic optimization (DO) for improved accuracy and real-time response to emerging threats. Almazroi[17] utilizes EffiIncepNet, an ensemble DL method, for BC security and medical data classification. The aim is to develop an efficient, scalable, and robust method that reduces security attacks and functions well. EffiIncepNet enhances the accuracy of classification and effectiveness of execution by associating InceptionResNetv2 and EfficientNet structures. EffiIncepNet data classification, internal BC performance to terminate doubtful transactions, and continuous system monitoring with models that look for strange behaviours are each part of the 3-phase security framework.

Rani et al.[18] projected a sophisticated neural network and AdaHessian optimizer model for defence and cryptojacking prediction. This method gives an advanced DNN cryptojacking threat prediction method utilizing AdaHessian optimizer, pruning, and post-training quantization. A novel structure for rapid DNN training using AdaHessian optimizer can identify cryptojacking endeavours by reducing the cost of computation. Renukadevi et al.[19] propose the fusion of optimization with deep wavelet neural networks on the denial of wallet attack detection (FODWNN-DoWAD) methhodology. The mobile utilizes min-max-based normalization, pair barracuda swarm optimization (PBSO)-based optimal feature selection, deep wavelet neural network (DWNN)-based attack detection, and hierarchical learning-based chaotic crayfish optimizer (HLCCO)-based tuning. Ramos et al.[20] analyze how technical events like attacks, hard forks, and wallet breaches affect the financial performance of Proof-of-Work cryptocurrencies. The study also utilizes models such as the capital asset pricing model (CAPM), mean adjusted return (MAR), and arbitrage pricing theory (APT); the study quantifies abnormal returns to evaluate the economic effects of these cyber-attacks and inform regulatory decisions. Li et al.[21] developed FaaSMT. This framework enables real-time security checks and performance-cost optimization in serverless applications utilizing parallel log collection, Merkle tree (MT) algorithms, and heuristic optimization (HO). Joshi et al.[22] develop and provide a specialized dataset for detecting DoW attacks in serverless architectures, assisting in creating effective security solutions for containerized applications. Krishna et al.[23] explore advanced encryption and security techniques, including AI, BC, and compliance measures, to improve cloud computing data protection and threat detection. Mora et al.[24] create a realistic dataset simulating function calls to detect DoW attacks using ML and neural networks. Ranganatha and Mustafa[25] utilize a bidirectional 3d quasi-recurrent neural network (Bi-3DQRNN) model, with self-adaptive synthetic over-sampling technique (SASOS) for data balancing, enhanced artificial gorilla troops optimization (EAGTO) for parameter tuning, and Proof of Voting (PoV) consensus BC for secure transaction forecasting. Xiong et al.[26] expose and analyze the Warmonger attack, a DoS threat in serverless platforms, resulting from shared egress IPs being blocked due to malicious function behaviour. Musamih et al.[27] employ BC, composable NFTs, threshold cryptography, fully homomorphic encryption (FHE), and decentralized storage via IPFS. Lavi et al.[28] developed an extendable security threat detection model for serverless computing, utilizing native cloud monitoring tools to detect anomalous behaviour and compromised serverless functions. Table 1 summarizes the key findings and models on serverless safety threat detection.

Despite the crucial improvements in security solutions for serverless and BC environments, various limitations still exist. Existing models mainly concentrate on specific attack types, making them less adaptable across diverse threat scenarios. Several techniques still encounter difficulty effectively handling large-scale, real-time data processing, specifically in highly dynamic serverless platforms. Moreover, the lack of standardized, comprehensive datasets restricts the accuracy of anomaly detection systems. While few methods incorporate optimization and AI techniques, the efficiency of these techniques in complex, growing environments remains questionable. Additionally, most solutions fail to address the threats of minimizing false positives and ensuring system scalability. Lastly, integrating advanced cryptographic techniques in BC systems sometimes increases computational costs and complexity.

Ref.	Techniques	Metrics	Findings
11	BC, AI, LSTM, Smart Contracts	Monitoring Accuracy, Response Speed, Data Integrity, Anomaly Detection Rate	The system detects GNSS signal anomalies and improves monitoring accuracy and response time.
12	BC, Smart Contracts, DL-based IDS, Proof of Voting Consensus, IPS/IDS Modules	Accuracy, Precision, Recall, Sensitivity, Specificity, FPR, F1-Score, MCC	The framework mitigates detection/validation time and boosts attack prevention, making it ideal for real-time use.
13	PBAC, BC, Smart Contracts, IPFS	Access Control Efficiency, EHR Management Scalability, Smart Contract Execution Time	The proposed technique enhances EHR management efficiency and scalability.
14	SMOTE, ENN, RF, XGB, BA, Random and Bayesian Search	Accuracy, Error Rate, Computational Costs, Parameter Search Time	The BA enhanced performance, attaining higher accuracy and a lower error rate in EDoS attack detection.
15	AI, HE, SIMD, DNN, FedAvg Algorithm, Privacy-preserving Distributed Learning	Detection Accuracy, Training Time, Real-world Adaptability	The framework maintains accuracy, mitigates training time, and adapts to real-world systems.
16	Ensemble Learning, WCA, DO, Threat Detection	Accuracy, Resilience Against Threats, Real-time Detection	Improved threat detection and system adaptability utilizing ensemble learning and WCA.
17	EfficientNet, Inception-ResNet-v2, Ensemble DL, BC, Continuous network monitoring, Anomaly detection algorithms	Classification Accuracy, AUC, Training time, Balanced Accuracy Recall Weighted Score (BPRWS) Metric	EffiIncepNet achieved up to 98% accuracy with enhanced scalability and security.
18	CNN, AdaHessian Optimization, Model Pruning, Post-training Quantization	Accuracy, Recall, Precision, F1-Score	Attained the highest output with low resource usage and high cryptojacking detection.
19	FODWNN-DoWAD, PBSO, DWNN, HLCCO	Accuracy, Recall, Precision, F1-Score, MCC	Highlighted more significant results in detecting DoW attacks utilizing optimized neural networks.
20	Cryptocurrency market data and Event study analysis, Attack type evaluation	Market Returns, Cryptocurrency Price Stability	Diverse attacks significantly affect cryptocurrency returns and stability.
21	FaaSMT, Parallel Processing, MTA, HO	Attack Detection, Function Monitoring, Performance Overhead	FaaSMT effectively detects attacks while mitigating performance overhead.
22	Dataset Creation, DoW Attack Detection, Containerized Applications, Threat Understanding	Attack Detection, Dataset Utilization, Security Enhancement	The dataset assists in developing stronger models for detecting DoW attacks in serverless environments.
23	AI, BC, and Compliance Measures	Data Protection, Threat Detection, Security Integrity	Encryption, AI, and BC improve cloud security and ensure compliance.
24	Synthetic Data Generation, ML	Dataset Generation, Anomaly Detection	Created a synthetic dataset for detecting DoW attacks in serverless environments using ML.
25	Bi-3DQRNN, SASOS, EAGTO	Accuracy	The model attained higher accuracy than existing fraud detection techniques in mobile transactions.
26	Warmonger Attack, Egress IP Analysis	Egress IP Usage Patterns, Number of Egress IPs per Serverless Service Providers (SSP)	Small egress IP sets in serverless platforms allow malevolent users to trigger IP blocking, causing DoS.
27	BC, NFTs, Threshold Cryptography, FHE, IPFS, Smart Contracts, Composable NFTs	Privacy Preservation, Data Traceability, Cost Evaluation, Feasibility and User-friendliness	The model ensures efficient, private, traceable genomic data management and monetization.
28	Native Monitoring Tools	Attack Detection Rate, False Alarm Rate	The methodology detects overall simulated attacks with minimal false alarms.

Table 1. Summary of key findings and techniques of studies on serverless security threat detection.

Materials and methods

This study proposes the MMDoWA-ARDL approach. The primary purpose of the model effectively detects and mitigates malicious attacks in serverless environments using an advanced DL model. Figure 1 depicts the workflow of the MMDoWA-ARDL model.

Z-score normalization

At first, the presented MMDoWA-ARDL model applies data pre-processing using Z-score normalization to transform input data into a helpful format[29]. This is chosen for its effectualness in standardizing data by transforming features with a mean of 0 and a standard deviation of 1. The model confirms that the overall input factors play a crucial role in equally contribution to the learning process, averting bias produced by scale differences. Unlike min-max normalization, which can be sensitive to outliers, Z-score is more robust in datasets with varying distributions. It also enhances the stability and convergence speed of optimization approaches. Moreover, it improves model generalization by mitigating the impact of noisy or extreme values. These merits make Z-score normalization an ideal choice for pre-processing in ML processes. At first, this is used to measure input data. This method is selected for data pre-processing owing to its efficacy in normalizing features. It converts them into a general measure with $an\ SD$ of 1 and a mean of 0. This model is mainly beneficial when managing databases that display fluctuating measures and allocations. This standardization safeguards the shape, which makes it suitable for techniques that pretend to spread data typically. Furthermore, this approach enables quicker convergence throughout the optimizer, as it helps uphold a constant input value, eventually improving the model's performance. Its efficiency and simplicity in enhancing comparability and understandability through features uphold its extensive usage in the applications of ML. Thus, this normalization is a statistical method for regularizing the features of the data. It transforms the data into an SD (σ) of 1 and a mean (μ) of 0. The mathematical formulation is shown as follows:

$$z=\frac{(x-\mu)}{\sigma} \tag{1}$$

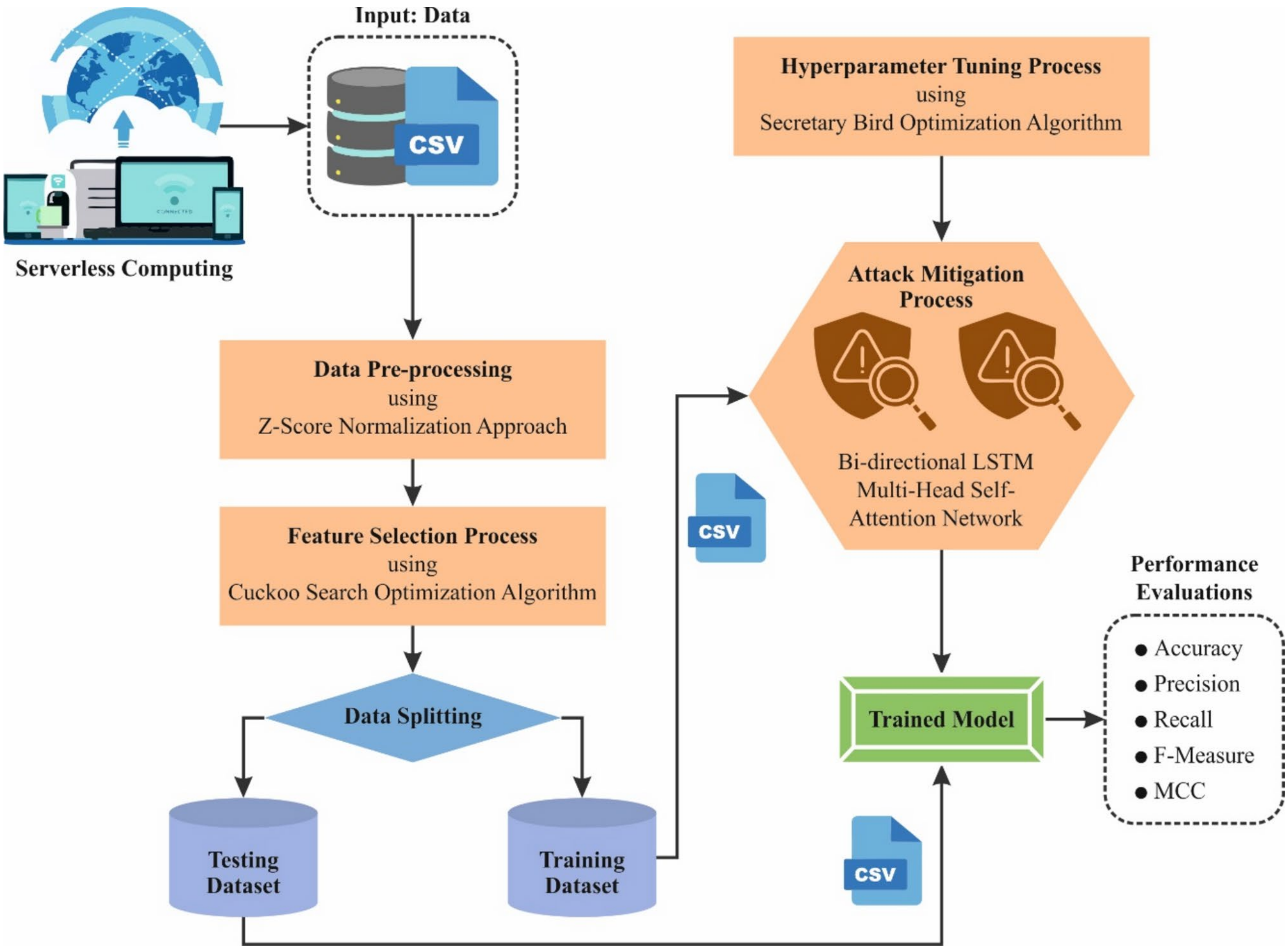

Input: Data
CSV
Serverless Computing
Data Pre-processing using Z-Score Normalization Approach
Feature Selection Process using Cuckoo Search Optimization Algorithm
Data Splitting
Testing Dataset
Training Dataset
Hyperparameter Tuning Process using Secretary Bird Optimization Algorithm
Attack Mitigation Process
Bi-directional LSTM Multi-Head Self-Attention Network
CSV
Trained Model
CSV
Performance Evaluations
● Accuracy
● Precision
● Recall
● F-Measure
● MCC

Fig. 1. Workflow of MMDoWA-ARDL approach.

While x denotes an original value, 0 displays the SD, and μ represents the mean of the feature.

CSO-based feature selection

Next, the FS process-based CSO efficiently identifies the most impactful attributes related to potential malicious activity[30]. This model is chosen for to its robust global search capability and simplicity of implementation. Inspired by the brood parasitism behaviour of cuckoos, CSO effectually explores the search space to detect the most relevant features while averting local minima. Compared to conventional methods such as genetic algorithms (GAs) or particle swarm optimization (PSO), CSO gives improved convergence rates and requires fewer parameter adjustments. Its use of Lévy flights improves exploration, making it appropriate for high-dimensional and complex datasets. By choosing optimal features, CSO mitigates computational costs and enhances the accuracy and interpretability of the model. Figure 2 portrays the working flow of the CSO methodology.

The features were recognized using the CSO technique from the set of features. This method is dependent upon cuckoo types, which perform resource exploitation by egg-laying in other crowded birds' nests. The host bird will also throw the eggs away as they do not fit into the nest or will build a novel nest by deserting the present one. Regarding the computational procedure, every egg in the nest signifies the solution, where the precise cuckoo egg maps to the novel solution. The cuckoo technique selects features by repeatedly hunting for an optimum subset of features. It develops the searching space, assesses the fitness of every feature sub-set, and upgrades the solutions depending upon the behaviour of the cuckoo that aims to enhance accuracy, decrease overfitting, and improve efficacy. A better solution is achieved by substituting the old cuckoo egg with a novel one that fits well in the nest. The instructions behind CSO are denoted below:

- At a time, a cuckoo rests only one egg, thrown in any chosen nest randomly.
- The fine eggs in the best nest are delivered to the next generation.
- Only a fixed amount of host nests are obtainable, and the likelihood of discovering a foreign egg is set within the range of [0,1]. With this likelihood, the egg is dropped by the host bird or dumped and transferred into a newly made nest in another position.

Depending upon the rules above, the cuckoo search is prolonged as below,

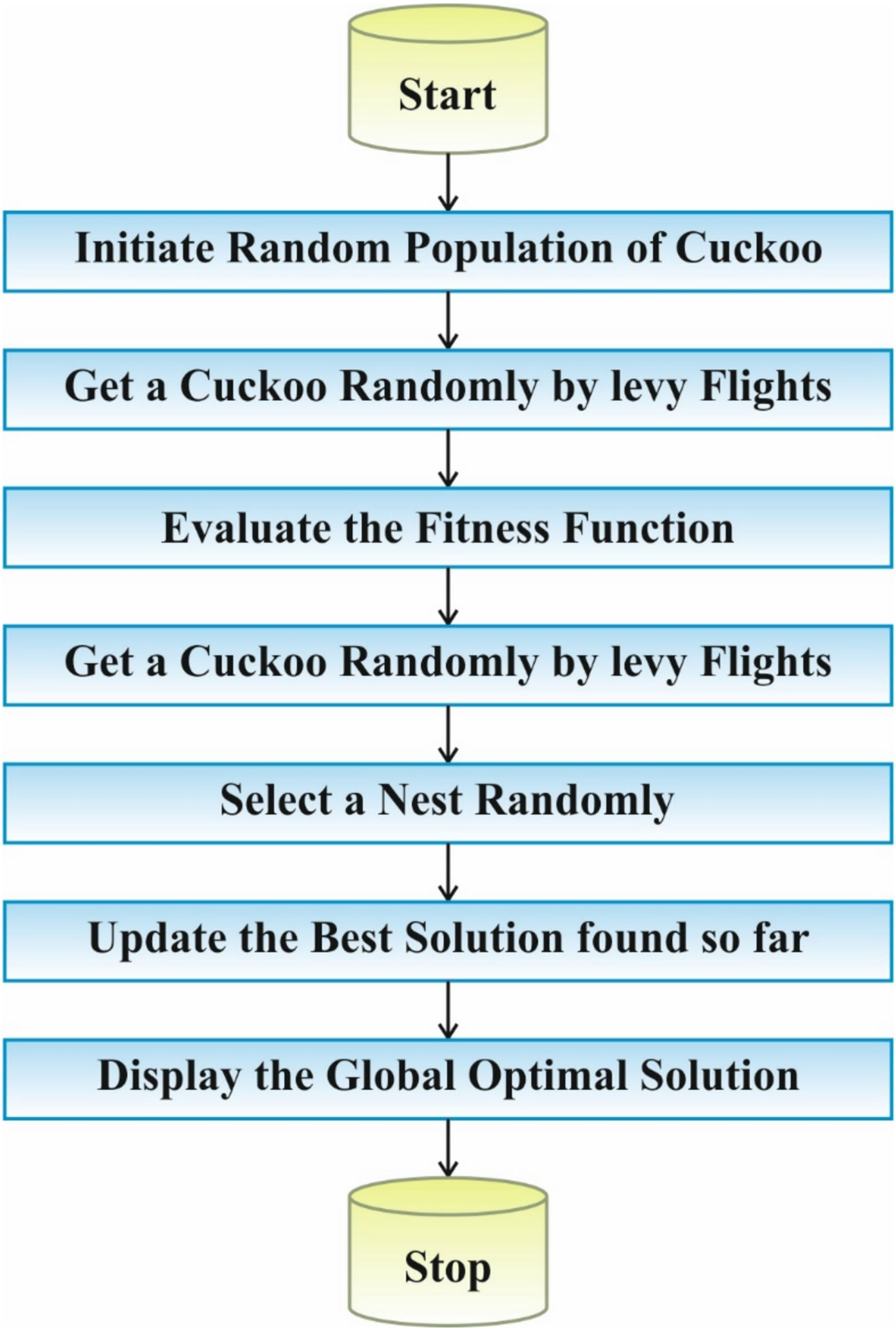

Start
Initiate Random Population of Cuckoo
Get a Cuckoo Randomly by levy Flights
Evaluate the Fitness Function
Get a Cuckoo Randomly by levy Flights
Select a Nest Randomly
Update the Best Solution found so far
Display the Global Optimal Solution
Stop

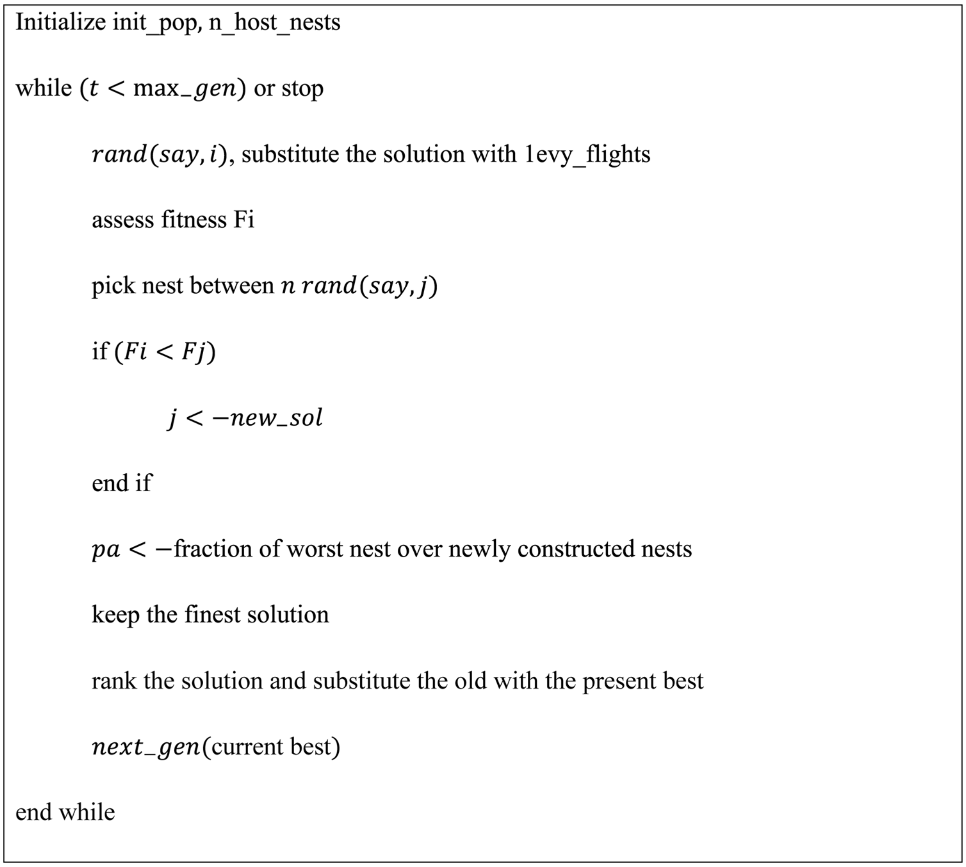
Fig. 2. Working flow of CSO technique.

Throughout the procedure of making solutions, the novel generations $x(t+1)$, for example, Levy flight, is formulated as

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus Le\nu\, y\left(\lambda\right) \tag{2}$$

In the abovementioned formulation, α denotes the size of size, which differs based on the issue. Frequently, the α value is equivalent to 1. The $\mathcal{B}$ symbol means the entry-wise multiplications. The Levy flight delivers a vital random walk, while the length of the step is drawn at random from the distribution of Levy.

$$Levy \sim\ u = t^{-\lambda},\ (1 < \lambda\ < 4) \tag{3}$$



```
Initialize init_pop, n_host_nests
while ($t <$ max_$gen$) or stop
    $rand(say, i)$, substitute the solution with 1evy_flights
    assess fitness Fi
    pick nest between $n\ rand(say, j)$
    if ($Fi < Fj$)
        $j < -new\_sol$
    end if
    $pa < -$fraction of worst nest over newly constructed nests
    keep the finest solution
    rank the solution and substitute the old with the present best
    $next\_gen$(current best)
end while
```

Algorithm 1. Pseudocode of CSO.

In the CSO model, the purposes are united into a single objective formulation, and a present weight categorizes every objective's significance. An FF that unites both FSs is used, as shown in Eq. (4).

$$Fitness\left(X\right) = \alpha\ \cdot\ E\left(X\right) + \beta\ \left(1 - \frac{|R|}{|N|}\right) \tag{4}$$

Here, $Fitness\left(X\right)$ is the fitness value of a sub-set X, $E\left(X\right)$ represents the classifier rate of error in the X subset, $|N|$ and $|R|$ means a quantity of original features and amount of chosen features in the dataset correspondingly, β and α means a decrease ratio and weights of classifier error, $\alpha\ \in\ [0,1]$ and $\beta\ = (1 - \alpha\ )$.

DoW attack mitigation process using BMNet

For the attack mitigation process, the BMNet model is utilized[31]. This approach is chosen because it captures transaction data's sequential patterns and contextual dependencies. By incorporating Bi-LSTM and multi-head self-attention, BMNet effectually learns long-term dependencies and concentrates on crucial aspects in past and future contexts. This dual capability outperforms conventional RNNs or CNNs, which may face difficulty with temporal dynamics or attention mechanisms. BMNet model improves detection accuracy in complex, growing attack scenarios and gives robustness against subtle anomalies. Its architecture confirms high sensitivity and adaptability, making it ideal for securing BC-based wallets. Figure 3 portrays the BMNet architecture.

The Bi-LSTM method develops time-based dependencies, exploiting its capability to handle time-based data bi-directionally. BiLSTM concurrently handles data at every time step over both backward and forward layers.

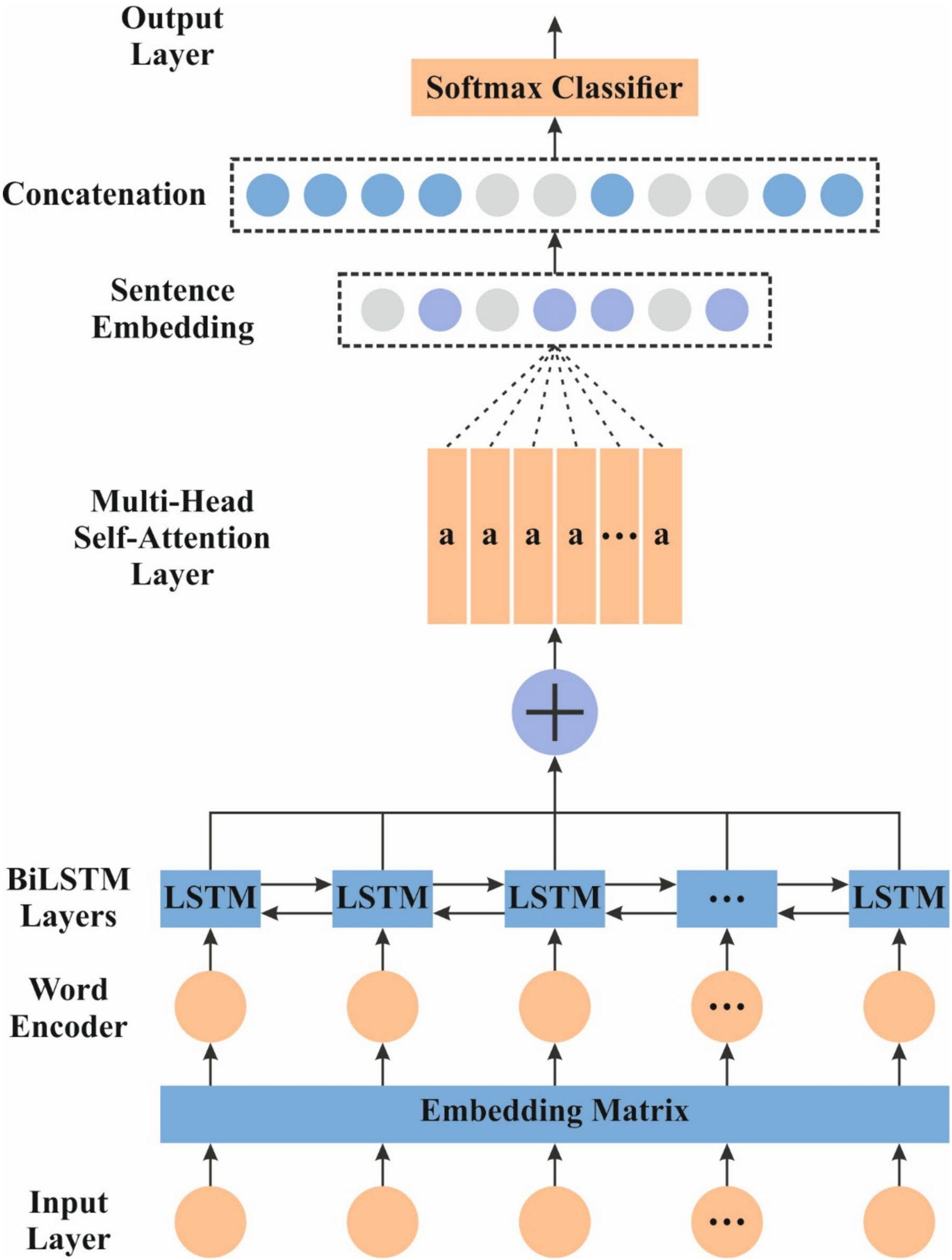


Fig. 3. Structure of BMNet model.

For every input L_t, the Bi-LSTM produces backward and forward hidden layers (HLs) for every time step, which are then linked to yield a feature representation h_t that creates both backward and forward time-based relationships:

$$h_t = BiLSTM\left(L_t\right), \; h_t \in \mathbb{R}^d \tag{5}$$

While, h_t signifies the temporal feature of tth, and d means a dimension of HL. The Bi-LSTM method makes a complete time-series feature matrix H by doing backwards and forward calculations on the complete sequence:

$$H = [h_1,\ h_2,\ \dots\ ,\ h_T]\,,\ H \in\ \mathbb{R}^{T\times\ d} \tag{6}$$

Here, T signifies the number of features. This feature matrix efficiently considers time-based dependencies, delivering more informative inputs for the following self-attention mechanism.

Afterwards, the time-based features made by the Bi-LSTM are distributed to the multi-head self-attention mechanism (MHSA) to improve the spatiotemporal feature-capturing ability by utilizing manifold attention heads to emphasize the global relations among dissimilar landmark features simultaneously. The MHSA aids in recognizing anomalies in local features from the spatio-temporal dimension and holds global data of the temporal sequence.

In BMNet, an input to the MHSA component is the time-series feature matrix H handled by the Bi-LSTM, which efficiently shows the global dependencies. It focuses on local feature variations and also captures temporal consistency. For every attention head, the key K, query Q, and value V are made utilizing weighting matrices W_Q, W_K, and W_V. The last attention output, Z, is gained by computing the attention score A:

$$Q = W_Q H,\ \ K = W_K H,\ \ V = W_V H \tag{7}$$

$$A = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) \tag{8}$$

$$Z = AV \tag{9}$$

The last MHSA output is gained by calculating the outputs of manifold attention heads Z_i in analogous:

$$Z_{MHSA} = Concat\,(Z_1,\ Z_2,\ \dots\ ,\ Z_h) \tag{10}$$

Here, h denotes the number of attention heads, and Z_{MHSA} is the output feature of MHSA.

SBOA-based parameter tuning

Finally, the hyper-parameter tuning model is implemented by SBOA to enhance the classification outcomes of the BMNet model. This method is chosen due to its dynamic balance between exploration and exploitation. Inspired by the hunting strategy of secretary birds, SBOA effectively searches the hyperparameter space to find optimal configurations with lesser iterations. SBOA adapts better to complex, non-linear problem spaces than conventional grid or random search methods. It mitigates the chances of getting stuck in local optima and accelerates convergence. Its lightweight computation and robust global search abilities make it ideal for tuning DL models, ultimately improving performance and stability. Table 2 describes the hyperparameters of CSO and SBOA models. Figure 4 illustrates the steps comprised in the SBOA approach.

The SBOA model is a population-based metaheuristic technique that makes a set of candidate solutions at random in the search space[32].

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,j} & \cdots & x_{1,D} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,j} & \cdots & x_{2,D} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & x_{i,2} & \cdots & x_{i,j} & \cdots & x_{i,D} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,j} & \cdots & x_{N,D} \end{bmatrix}_{N\times D} \tag{11}$$

Algorithm	Hyperparameter	Description	Typical Value/Range
CSO	NO_of_Nests	Overall candidate solutions (POPULACE_SIZE)	15–50
	DISCOVERY_Rate_of_Alien_Eggs (p.a.)	Probability of finding a bad solution and replacing it	0.25–0.35
	MAX_Iterations	MAX_NO_of_Generations	100–1000
	SIZE_of_Step (α)	Regulates the scale of random walk (Lévy flight)	0.01–1
SBOA	POPULACE_Size	OVERALL_Search_Agents	20–50
	MAX_Iterations	NO_of_OPTZ_Cycles	100–1000
	ESCAPE_Energy (E)	Regulates the behaviour of exploration vs. exploitation	0 to 1 (adaptive)
	FLIGHT_Angle_Factor (θ)	Governs the variability of the search direction	π/6 to π/3 (radians)
	CONVERGENCE_Coefficient (C)	Adjusts the intensity of dislocation towards prey	0.5–2

Table 2. Hyperparameters and their typical values for CSO and SBOA, utilized to control exploration, convergence, and overall optimization performance.

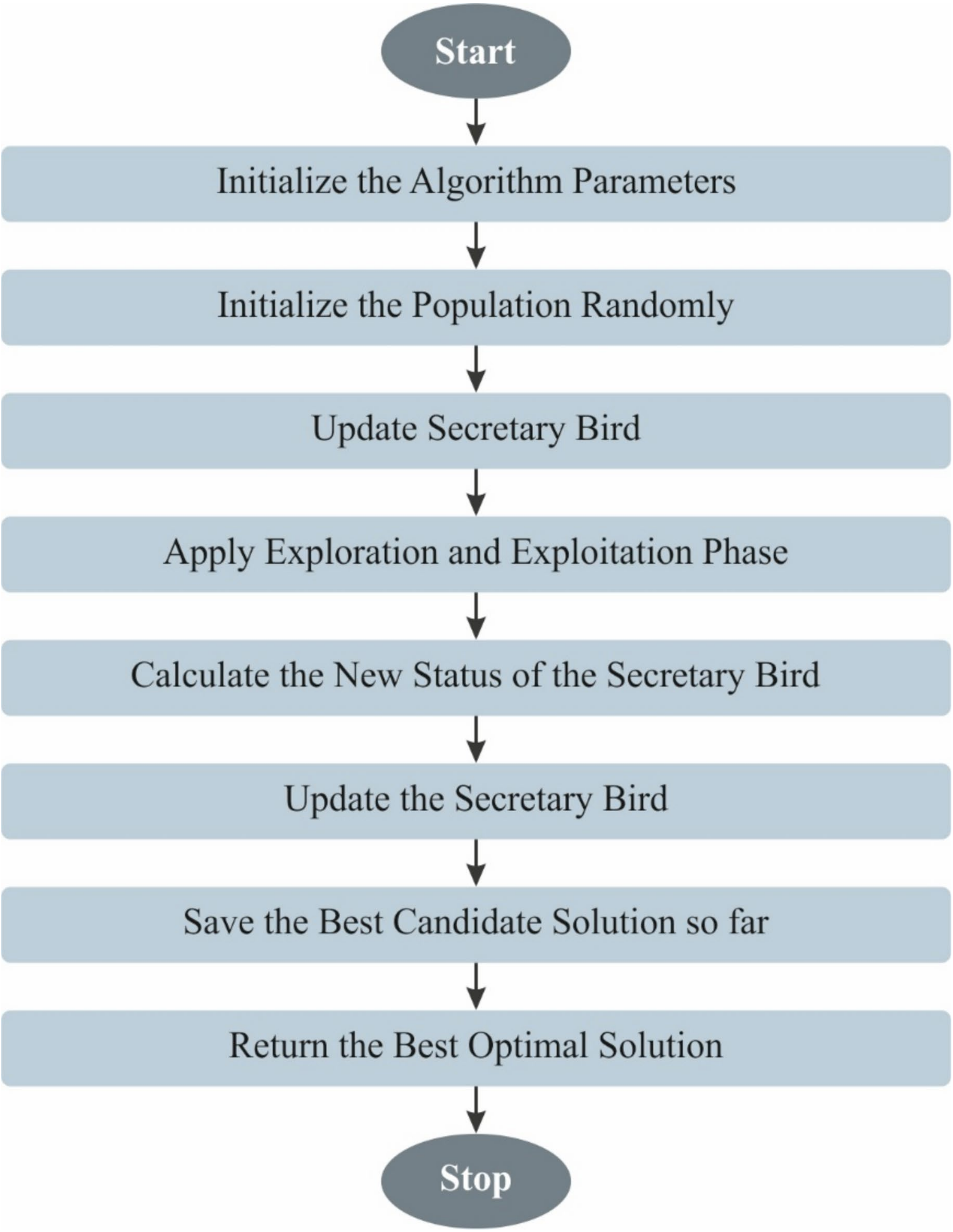

Start
Initialize the Algorithm Parameters
Initialize the Population Randomly
Update Secretary Bird
Apply Exploration and Exploitation Phase
Calculate the New Status of the Secretary Bird
Update the Secretary Bird
Save the Best Candidate Solution so far
Return the Best Optimal Solution
Stop

Fig. 4. Steps involved in the SBOA method.

Here, X signifies the population of secretary birds, X_i means the location of $the\ i^{th}$ secretary bird, $x_{i,j}$ specifies the location information of the j^{th} problem variable for i^{th} secretary birds, $and\ N$ represents the population dimension. D refers to the dimension of problem variables.

The Secretary Birds initial locations are defined at random depend upon Eq. (12):

$$x_{i,j} = (ub_j - lb_j) \times\ r_1 + lb_j \tag{12}$$

While, $x_{i,j}$ signifies the initial value of decision variable j for i^{th} candidate solutions; ub_j and lb_j refer to the upper and lower boundaries, correspondingly; and r_1 denotes an arbitrarily created number within the interval of $(0{,}1)$.

Hunting behaviour (exploration)

This kind of behaviour naturally contains 3 phases: prey searching (P_1), prey exhausting (P_2), and prey attacking (P_3). In the prey search phase, Secretary Birds are probable prey. If the prey is recognized, then they arrive at the prey-exhausting phase. With the firm decision of the prey's actions, the Secretary Birds easily fly, jump, and aggravate near the snake, exhausting the enemy's energy. Once the stamina of the prey is completely exhausted, then they will start an attack. This procedure was demonstrated utilizing Eqs. (13) and (14).

$$x_{i,j}^{new1} = \begin{cases} P_1\ :\ x_{i,j} + r_2 \times\ (x_{r_1} - x_{r_2})\ , & if\ iter < \frac{1}{3}T \\ P_2 : x_{best} + exp\left(\left(\frac{iter}{T}\right)^4\right) \times\ (RB - 0.5) \times\ (x_{best} - x_{i,j})\ , & if\ \frac{1}{3}T < iter < \frac{2}{3}T \\ P_3 : x_{best} + \left(1 - \frac{iter}{T}\right)^{\left(2\times\ \frac{iter}{T}\right)} \times\ x_{i,j} \times\ RL, & else \end{cases} \tag{13}$$

$$X_i \left\{ \begin{array}{cc} X_i^{new1}, & if\ F_i^{new1} < F_i \\ X_i, & else \end{array} \right. \tag{14}$$

While, $iter$ means the present count of iteration, T embodies the maximum iteration count, X_i^{new1} specifies the novel state of i^{th} secretary bird in the 1st phase, and x_{r1} and x_{r2} are candidate solutions at random for the 1st phase iteration. r_2 represents a generated array at random dimension $1\text{x}\,D$ within the range of $[0{,}1]$. $x_{i,j}^{new1}$ means the location data of its j^{th} dimension, while F_i^{new1} represents its objective function fitness value. RB implies a range of sizes $1 \times\ D$ produced randomly, and X_{best} signifies the finest solution attained until now. RL denotes the function of Lévy flight, which is computed utilizing Eq. (15).

$$\begin{cases} RL = 0.5 \times\ Levy\,(Dim) \\ Levy\,(Dim) = 0.01 \times\ \frac{u\times\ \sigma}{|v|^{\frac{1}{\eta}}} \\ \sigma\ = \left(\frac{\varGamma\ (1+\eta\ )\times\ \sin\left(\frac{\pi\ \eta}{2}\right)}{\varGamma\ \left(\frac{1+\eta}{2}\right)\times\ \eta\ \times\ 2^{\left(\frac{\eta\ -1}{2}\right)}}\right)^{\frac{1}{\eta}} \end{cases} \tag{15}$$

While η represents a fixed constant with a value of 1.5, u and v denote produced numbers randomly in the range of $[0,\ 1]$, and $\varGamma$ embodies the gamma function.

Escape strategy (exploitation)

Secretary birds might tackle attacks from predators or attempts to catch its prey. Due to its cleverness, they frequently use avoidance tactics to defend themselves. These tactics were mainly separated into dual kinds: one includes soaring or running to escape (S), whereas others involve utilizing ecological colours or constitute camouflage (S), which makes it firmer for predators to perceive them. This procedure is demonstrated utilizing Eqs. (16) and (17).

$$x_{i,j}^{new2} = \left\{ \begin{array}{ll} S_1 : x_{best} + (2 \times\ RB - 1) \times\ \left(1 - \frac{iter}{T}\right)^2 \times\ x_{i,j}, & if\ q < r_3 \\ S_2 : x_{i,j} + r_4 \times\ (x_{rand} - x_{i,j})\ , & else \end{array} \right. \tag{16}$$

$$X_i = \left\{ \begin{array}{l} X_i^{new2}, if\ F_i^{new2} < F_i \\ X_i,\ else \end{array} \right. \tag{17}$$

In the above equation, $q = 0.5$, r_3, and r_4 signify sets of dimensions $(1 \times\ D)$ produced arbitrarily from a standard distribution. x_{rand} means a produced candidate solution randomly in the present iteration, and 1 denotes a selected number at random of either 1 or 2.

Fitness choice is a crucial factor in manipulating SBOA performance. The hyper-parameter range procedure contains the solution-encoded system for appraising the effectiveness of the candidate solution. The SBOA reflects accuracy as the foremost measure to project the fitness function. It is shown below:

$$Fitness\ =\ \max\ (P) \tag{18}$$

$$P = \frac{TP}{TP + FP} \tag{19}$$

Here, TP and FP signifies the true and positive values.

Experimental analysis

The performance evaluation of the MMDoWA-ARDL methodology is examined under a benchmark dataset[33]. The dataset covers 100,000 samples below dual classes, with every class comprising 50,000 samples, as defined in Table 3. It includes 17 features, out of which 14 are selected. Table 4 outlines key metrics utilized to detect DoW behaviour in serverless function invocations.

Transactions	No. of transactions
Attack transactions	50,000
Legitimate transactions	50,000
Total transactions	100,000

Table 3. Details of the dataset.

Feature	Description	Relevance to DoS/DDoS attacks
ID	Unique identifier for every entry.	Assists in tracking particular requests and analyzing attack patterns.
IP	SOURCE_IP address.	Utilized to detect the origin of attack requests. The high frequency may exhibit bot activity.
Bot	FLAG_if_IP_is_Bot (TRUE/FALSE).	The presence of a Bot is a key indicator of automated attack traffic.
FunctionId	Identifier of the specific function being triggered.	Function call patterns can assist in detecting unusual requests and illustrating an attack.
FunctionTrigger	FUNCTION_Trigger (e.g., notification).	Malicious activity may be the result of anomalous function triggers.
Timestamp	TIMESTAMP_Request.	It assists in detecting the time of attack and correlates with high traffic spikes.
SubmitTime	TIME_to_Submit a request.	Longer submission times may hint at attack attempts like flooding.
Round-Trip Time (RTT)	TIME_for_Signal to travel to the destination and back.	High RTT values may show network congestion due to an attack.
InvocationDelay	DELAY_before_Function_Invoke	Enhanced delays may suggest throttling from attack traffic.
ResponseDelay	The time between getting the request and sending a response.	Delays in responses show resource saturation, which is usual in DoS/DDoS.
FunctionDuration	DURATION_Function_Runs.	Long durations reflect attacks that overload system functions.
ActiveFunctionsAtRequest	ACTIVEFUNCTIONS_during_Request.	Higher numbers could indicate system stress from attack traffic.
ActiveFunctionsAtResponse	Number of active functions at the time of response.	A higher number may indicate overloading, revealing DoS attacks.
MaxCPU	MAX_CPU_USAGE during the request.	Enhanced CPU usage may show resource exhaustion from an attack.
AvgCPU	AVG_CPU_USAGE during the request.	Higher average CPU usage can illustrate a DoS/DDoS attack.
P95MaxCPU	The 95th percentile of maximum CPU usage.	It assists in highlighting outliers in CPU usage and helps detect spikes caused by attacks.
VMCategory	Category of virtual machine (e.g., Delay-insensitive).	VM classes assist in correlating attack types, such as delay-sensitive traffic overload.
VMCoreCountBucket	CPU_No. cores in the VM bucketed into categories.	Unusual core usage patterns may depict resource hogging from attack traffic.
VMMemoryBucket	Bucket for VM memory allocation.	Memory usage spikes may show resource exhaustion during an attack.

Table 4. Summary of features relevant to the detection of DoW attacks.

Figure 5 established a set of confusion matrices created by the MMDoWA-ARDL technique on several epochs. On 500 epochs, the MMDoWA-ARDL technique has accepted 49,477 samples into attack transactions and 49,712 samples into legitimate transactions. Followed by, on 1000 epochs, the MMDoWA-ARDL methodology has known 49,515 samples into attack transactions and 49,707 samples into legitimate transactions. Simultaneously, on 1500 epochs, the MMDoWA-ARDL methodology has accepted 49,536 samples into attack transactions and 49,699 samples into legitimate transactions. Moreover, in the 2000 epochs, the MMDoWA-ARDL approach has known 49,583 samples for attack transactions and 49,706 samples for legitimate transactions. Eventually, on 3000 epochs, the MMDoWA-ARDL technique has realized 49,659 samples into attack transactions and 49,730 samples into legitimate transactions.

In Table 5; Fig. 6, the DoW attack recognition results of the MMDoWA-ARDL methodology are shown below in various epochs. The outcomes identified that the MMDoWA-ARDL methodology correctly recognized the attack and legitimate transaction samples. With 500 epochs, the MMDoWA-ARDL methodology attains an average $accu_y$ of 99.19%, $prec_n$ of 99.19%, $reca_l$ of 99.19%, $F_{measure}$ of 99.19%, MCC of 98.38%, and Kappa of 98.63%. Besides, with 1000 epochs, the MMDoWA-ARDL methodology attains an average $accu_y$ of 99.22%, $prec_n$ of 99.22%, $reca_l$ of 99.22%, $F_{measure}$ of 99.22%, MCC of 98.44%, and Kappa of 98.67%. At the same time, with 1500 epochs, the MMDoWA-ARDL methodology accomplishes an average $accu_y$ of 99.24%, $prec_n$ of 99.24%, $reca_l$ of 99.24%, $F_{measure}$ of 99.23%, MCC of 98.47%, and Kappa of 98.78%. Also, with 2000 epochs, the MMDoWA-ARDL model accomplishes an average $accu_y$ of 99.29%, $prec_n$ of 99.29%, $reca_l$ of 99.29%, $F_{measure}$ of 99.29%, MCC of 98.58%, and Kappa of 99.01%. At the same time, with 3000 epochs, the MMDoWA-ARDL model obtains an average $accu_y$ of 99.39%, $prec_n$ of 99.39%, $reca_l$ of 99.39%, $F_{measure}$ of 99.39%, MCC of 98.78%, and Kappa of 99.09%. These outputs highlight the consistent and robust performance of the MMDoWA-ARDL model across varying training durations.

Figure 7 illustrates the TRA $accu_y$(TRAAY) and validation $accu_y$(VLAAY) outcomes of the MMDoWA-ARDL technique. The $accu_y$ analysis is calculated under 0-3000 epochs. The figure highlights that the TRAAY and VLAAY analysis showed an increasing trend, which informed the capacity of the MMDoWA-ARDL technique with higher outcomes across diverse iterations. Similarly, the TRAAY and VLAAY exhibit close epochs, indicating minimal overfitting and confirming the robustness of the MMDoWA-ARDL model under unseen data.

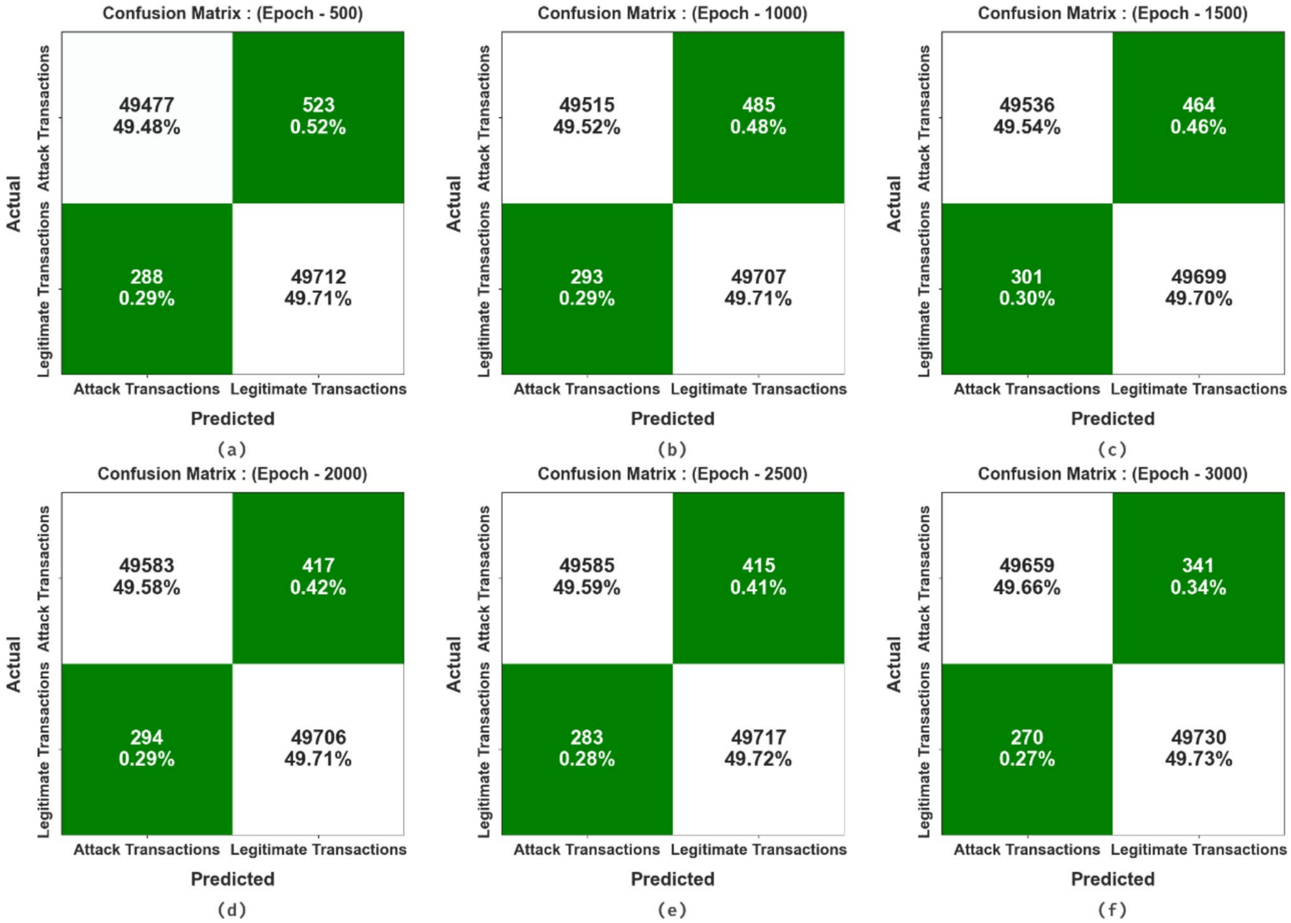


Fig. 5. Confusion Matrix of MMDoWA-ARDL approach (**a**–**f**) Epochs 500–3000.

Figure 8 depicts the TRA loss (TRALO) and VLA loss (VLALO) curves of the MMDoWA-ARDL method. The loss is computed across an interval of 0-3000 epochs. It is also shown that the TRALO and VLALO analysis establishes a diminishing algorithm, which informed the capability of the MMDoWA-ARDL technique to balance a trade-off between data fitting and generalization. There is a continuous reduction in loss, besides assurances of the more excellent performance of the MMDoWA-ARDL technique and tuning of the prediction outputs over time.

In Fig. 9, the precision-recall (PR) graph results of the MMDoWA-ARDL methodology below several epochs deliver clarification into its solution by plotting PR for each class. The steady increase in PR analysis between each class label describes the effectiveness of the MMDoWA-ARDL approach in the classification process.

Figure 10 inspects the ROC graph of the MMDoWA-ARDL methodology below several epochs. The results suggest that the MMDoWA-ARDL methodology gains more excellent ROC analysis across all classes, demonstrating an essential capacity for discerning classes. This constant trend of maximum ROC analysis across several classes means the capable outcomes of the MMDoWA-ARDL method on predicting class labels, highlighting the robust behaviour of the classification process.

To exhibit the proficiency of the MMDoWA-ARDL methodology, a thorough comparison investigation is conducted in Table 6; Fig. 11[20,34–37]. The table values depicted that the MMDoWA-ARDL methodology archives superior performance. Based on $accu_y$, the MMDoWA-ARDL methodology provides a better $accu_y$ of 99.39%. At the same time, NB, DBN, SVM, DQSP, DQN, DNN, Inception-ResNet, CAPM, MAR, and APT models have achieved lesser $accu_y$ values of 95.13%, 94.01%, 99.05%, 91.60%, 90.79%, 97.10%, 90.59%, 92.36%, 91.39%, and 97.88%, respectively. Afterwards, based on $prec_n$, the MMDoWA-ARDL approach presents a higher $prec_n$ of 99.39%, where NB, DBN, SVM, DQSP, DQN, DNN, Inception-ResNet, CAPM, MAR, and APT techniques have attained lower $prec_n$ values of 92.82%, 97.01%, 97.34%, 90.58%, 98.34%, 97.64%, 92.22%, 91.34%, 99.05%, and 98.24%, correspondingly. Simultaneously, depending on the $recal_l$, the MMDoWA-ARDL approach presents a maximum $recal_l$ of 99.39%. In contrast, Naïve Bayes (NB), DBN, SVM, DQSP, DQN, DNN, Inception-ResNet, CAPM, MAR, and APT methodologies have accomplished lower $recal_l$ values of 97.09%, 95.00%, 90.11%, 99.01%, 91.30%, 96.17%, 92.79%, 99.13%, 91.93%, and 96.94%, subsequently. Lastly, for $F_{measure}$, the MMDoWA-ARDL approach presents a superior $F_{measure}$ of 99.39%. In contrast, NB, DBN, SVM, DQSP,

Class Labels	$Accu_y$	$\mathbf{Prec_n}$	$\mathbf{Reca_l}$	$F_{Measure}$	MCC	Kappa
Epoch – 500						
Attack Transactions	98.95	99.42	98.95	99.19	98.38	98.55
Legitimate Transactions	99.42	98.96	99.42	99.19	98.38	98.71
Average	99.19	99.19	99.19	99.19	98.38	98.63
Epoch – 1000						
Attack Transactions	99.03	99.41	99.03	99.22	98.44	98.80
Legitimate Transactions	99.41	99.03	99.41	99.22	98.44	98.54
Average	99.22	99.22	99.22	99.22	98.44	98.67
Epoch – 1500						
Attack Transactions	99.07	99.40	99.07	99.23	98.47	99.21
Legitimate Transactions	99.40	99.08	99.40	99.24	98.47	98.34
Average	99.24	99.24	99.24	99.23	98.47	98.78
Epoch – 2000						
Attack Transactions	99.17	99.41	99.17	99.29	98.58	99.11
Legitimate Transactions	99.41	99.17	99.41	99.29	98.58	98.90
Average	99.29	99.29	99.29	99.29	98.58	99.01
Epoch – 2500						
Attack Transactions	99.17	99.43	99.17	99.30	98.60	99.00
Legitimate Transactions	99.43	99.17	99.43	99.30	98.60	99.12
Average	99.30	99.30	99.30	99.30	98.60	99.06
Epoch – 3000						
Attack Transactions	99.32	99.46	99.32	99.39	98.78	98.67
Legitimate Transactions	99.46	99.32	99.46	99.39	98.78	99.50
Average	99.39	99.39	99.39	99.39	98.78	99.09

Table 5. DoW attack detection of MMDoWA-ARDL method under various epochs.

DQN, DNN, Inception-ResNet, CAPM, MAR, and APT approaches have reached minimal $F_{measure}$ values of 94.99% 97.56%, 91.20%, 97.34%, 91.69%, 90.57%, 95.01%, 97.84%, 92.20%, and 91.26%, subsequently.

The computation time (CT) outcomes of the MMDoWA-ARDL approach are compared to other existing methods in Table 7; Fig. 12. The results signified that the MMDoWA-ARDL approach achieves an inferior CT of 5.72s. On the other hand, the NB, DBN, SVM, DQSP, DQN, DNN, Inception-ResNet, CAPM, MAR, and APT methods achieve increased CT values of 8.60s, 9.63s, 8.74s, 19.52s, 17.37s, 16.68s, 18.32s, 11.90s, 10.56s, and 12.25s, correspondingly. Thus, the MMDoWA-ARDL approach is used to mitigate malicious DoW attacks.

Table 8; Fig. 13 demonstrates the ablation study of the MMDoWA-ARDL approach. Four models namely CSO, SBOA, BMNet, and the proposed MMDoWA-ARDL model are analyzed under $accu_y$, $prec_n$, $reca_l$, $F_{measure}$, MCC, Kappa. The CSO and SBOA models illustrates competitive results, attaining $accu_y$ of 97.35% and 97.86%, respectively. BMNet additionally enhanced performance with an $accu_y$ of 98.65%. However, the proposed MMDoWA-ARDL model outperformed all baselines, attaining a consistent 99.39% across all metrics. These results confirm that the integration of attribute reduction and DL techniques in the MMDoWA-ARDL model significantly improves detection capabilities, validating its robustness and superiority in securing serverless applications against DoW attacks.

Conclusion

In this study, the MMDoWA-ARDL approach is proposed. The primary purpose of the MMDoWA-ARDL approach is to propose a novel framework that effectively detects and mitigates malicious attacks in serverless environments using an advanced DL model. At first, the presented MMDoWA-ARDL model applies data pre-processing using Z-score normalization to transform input data into a valid format. Furthermore, the FS process-based CSO efficiently identifies the most impactful attributes related to potential malicious activity. The BMNet model is employed for the attack mitigation process. Finally, SBOA accomplishes the parameter tuning method to enhance the classification outcomes of the BMNet model. A wide-ranging experimental investigation is conducted using a benchmark dataset to exhibit the superior performance of the proposed MMDoWA-ARDL technique. The comparison study of the MMDoWA-ARDL model portrayed a superior accuracy value of 99.39% over existing techniques.

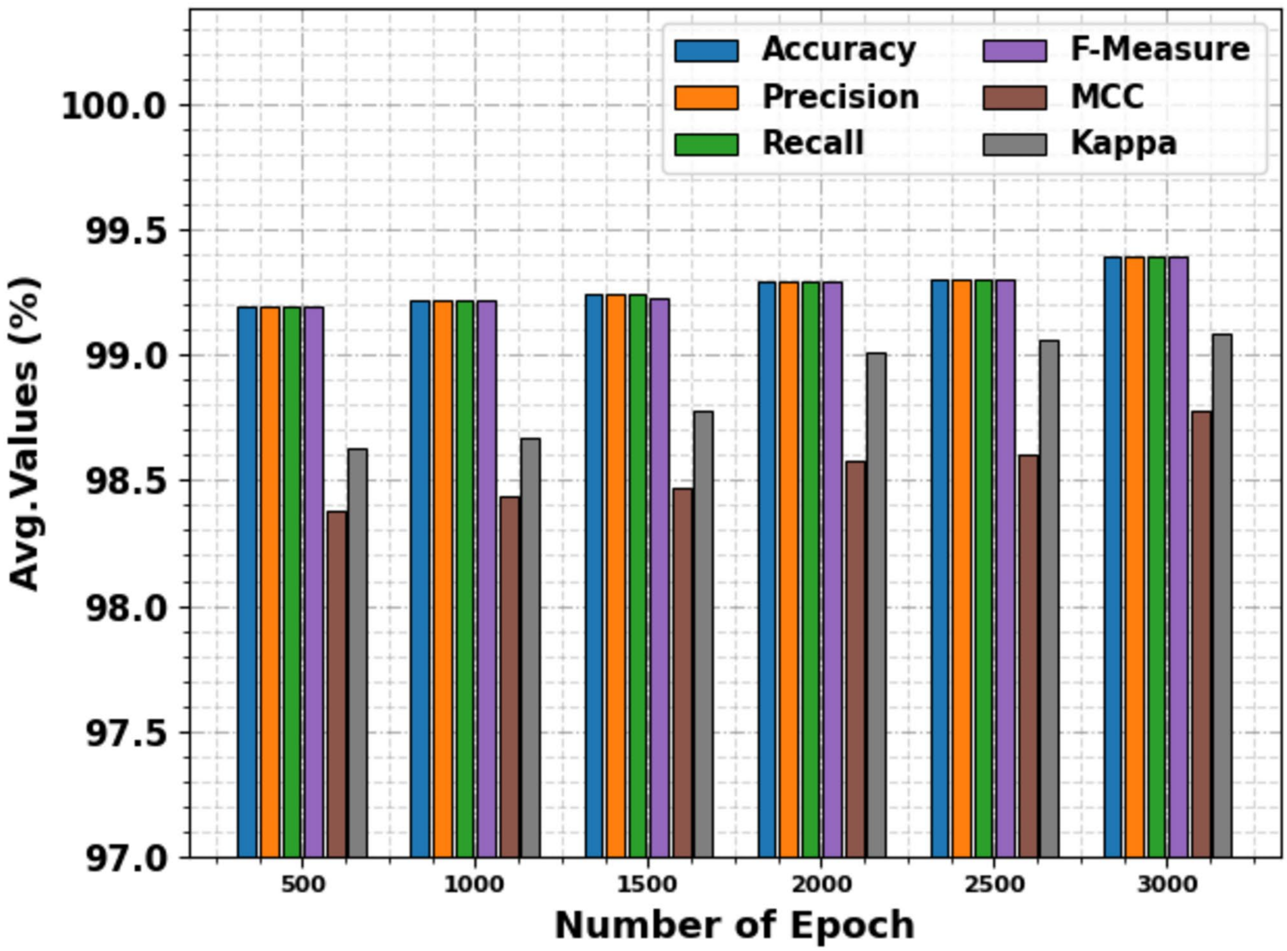


Fig. 6. Average result of MMDoWA-ARDL method below several epochs.

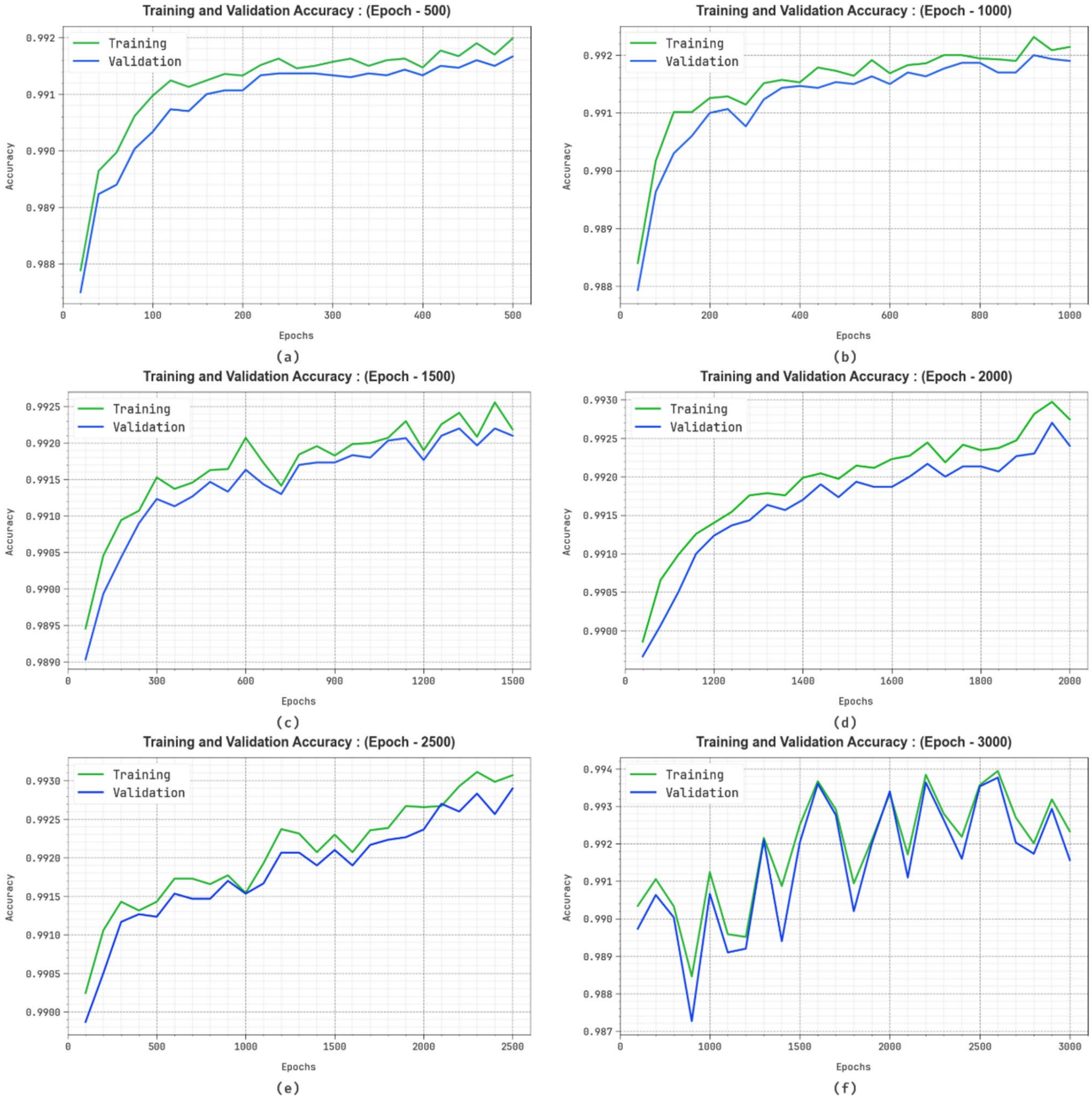


Fig. 7. $Accu_y$ curve outcome of MMDoWA-ARDL method (**a**–**f**) Epochs 500–3000

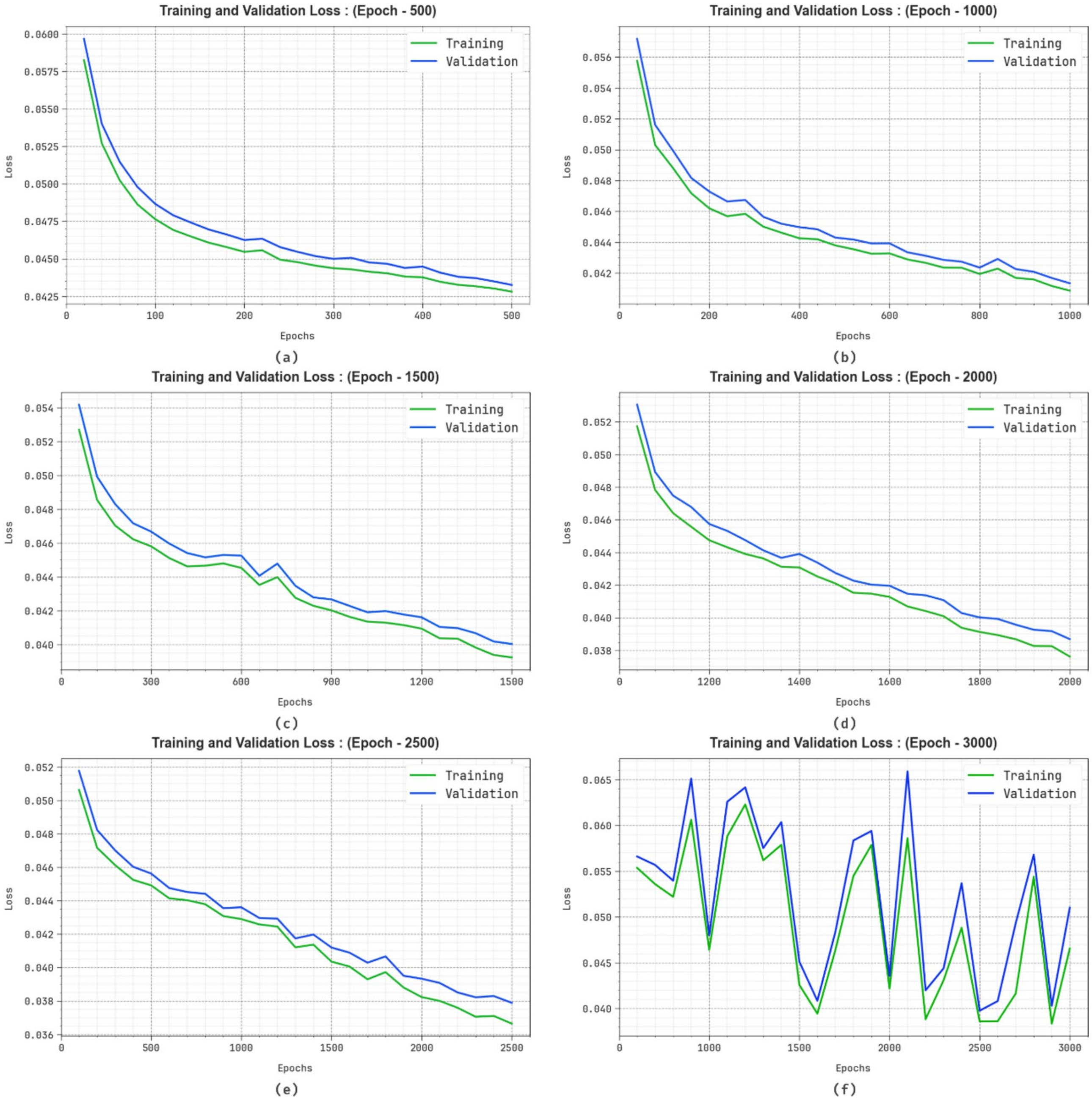


Fig. 8. Loss graph outcome of MMDoWA-ARDL approach (**a**–**f**) Epochs 500–3000.

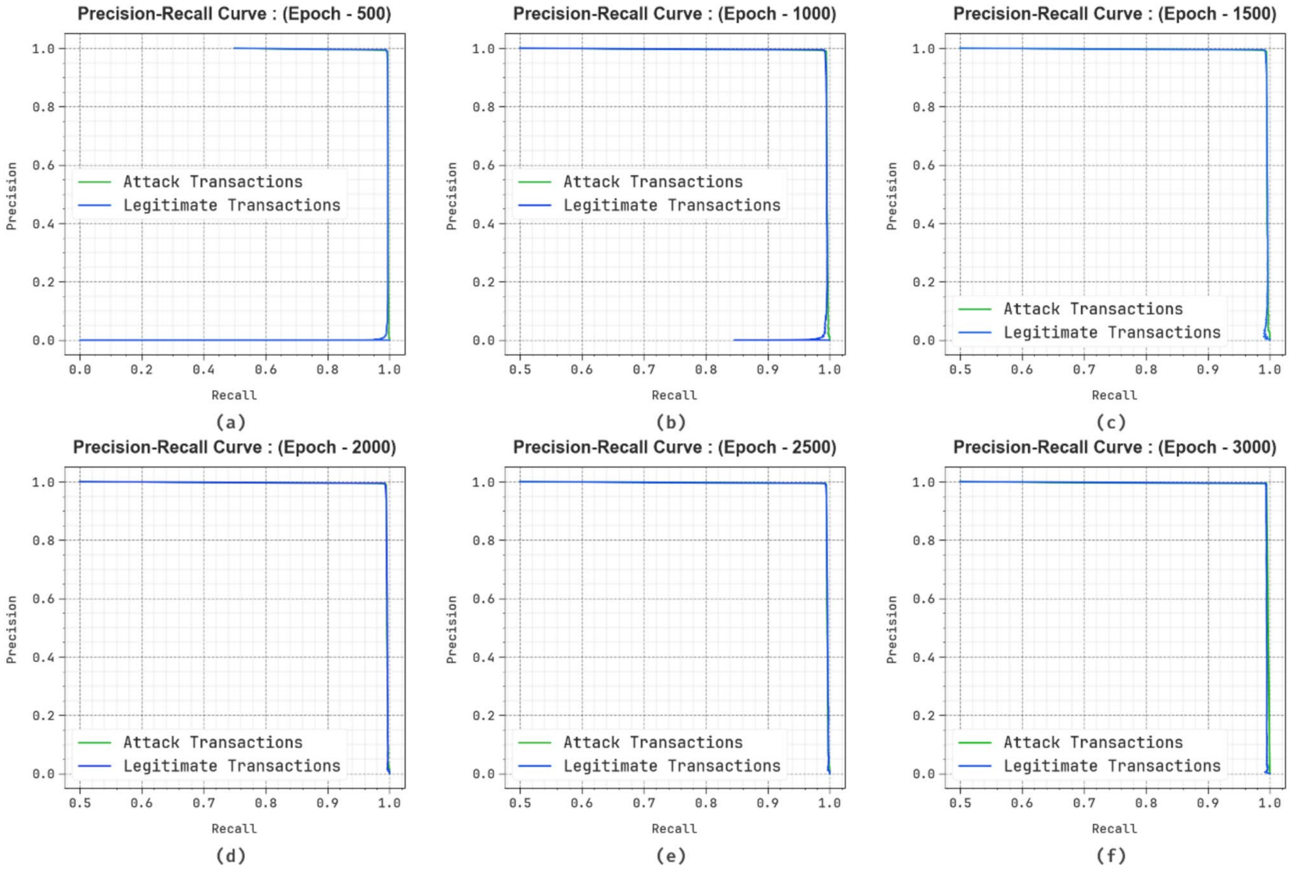


Fig. 9. PR curve outcome of MMDoWA-ARDL approach (**a**–**f**) Epochs 500–3000.

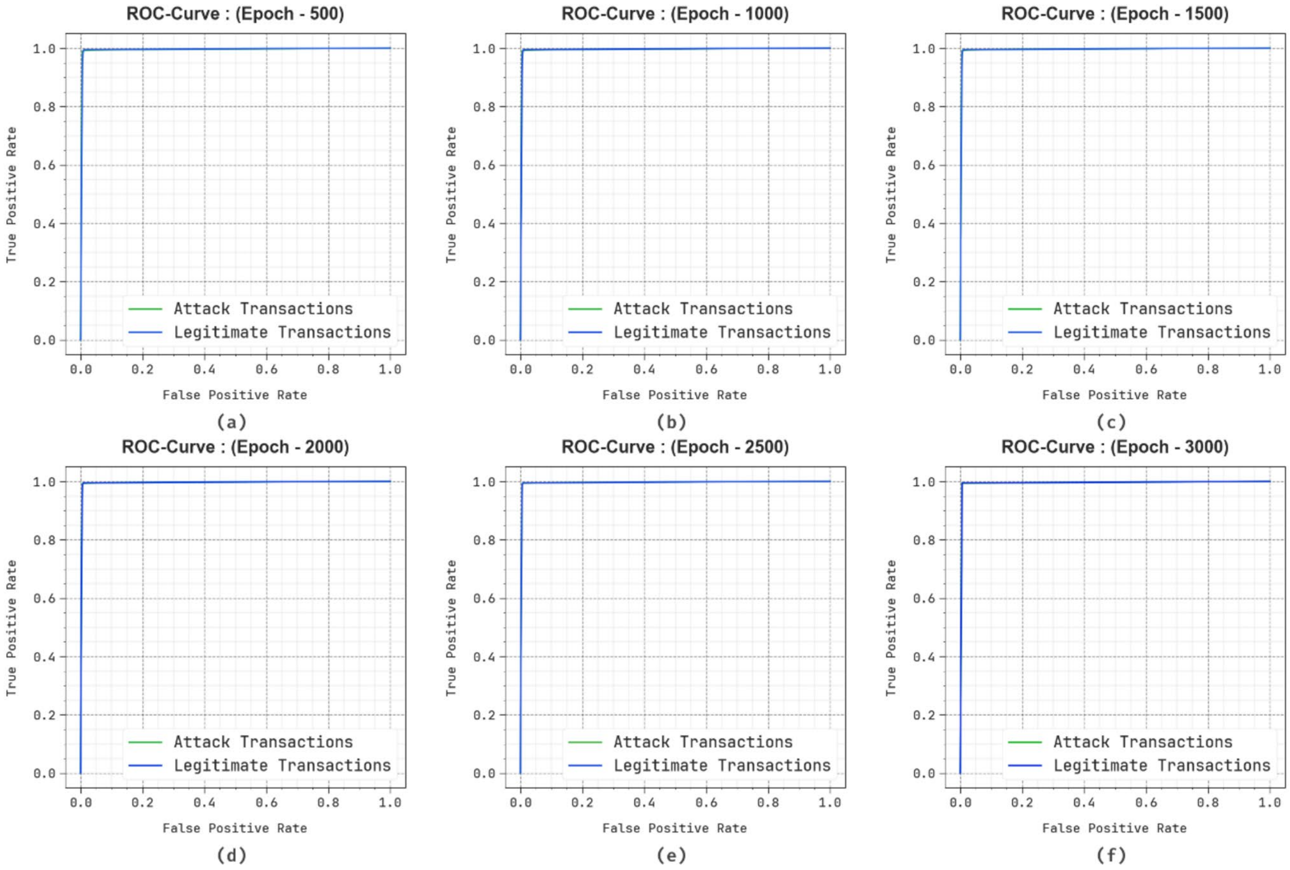


Fig. 10. ROC analysis outcome of MMDoWA-ARDL model (**a**–**f**) Epochs 500–3000.

Classifier	$Accu_y$	$\mathbf{Prec}_n$	$\mathbf{Reca}_l$	$F_{Measure}$
NB	95.13	92.82	97.09	94.99
DBN Model	94.01	97.01	95.00	97.56
SVM Method	99.05	97.34	90.11	91.20
DQSP Model	91.60	90.58	99.01	97.34
Deep Q-Network	90.79	98.34	91.30	91.69
DNN Algorithm	97.10	97.64	96.17	90.57
Inception-ResNet	90.59	92.22	92.79	95.01
CAPM	92.36	91.34	99.13	97.84
MAR	91.39	99.05	91.93	92.20
APT	97.88	98.24	96.94	91.26
MMDoWA-ARDL	99.39	99.39	99.39	99.39

Table 6. Comparative outcome of MMDoWA-ARDL approach with existing methods[20,34–37].

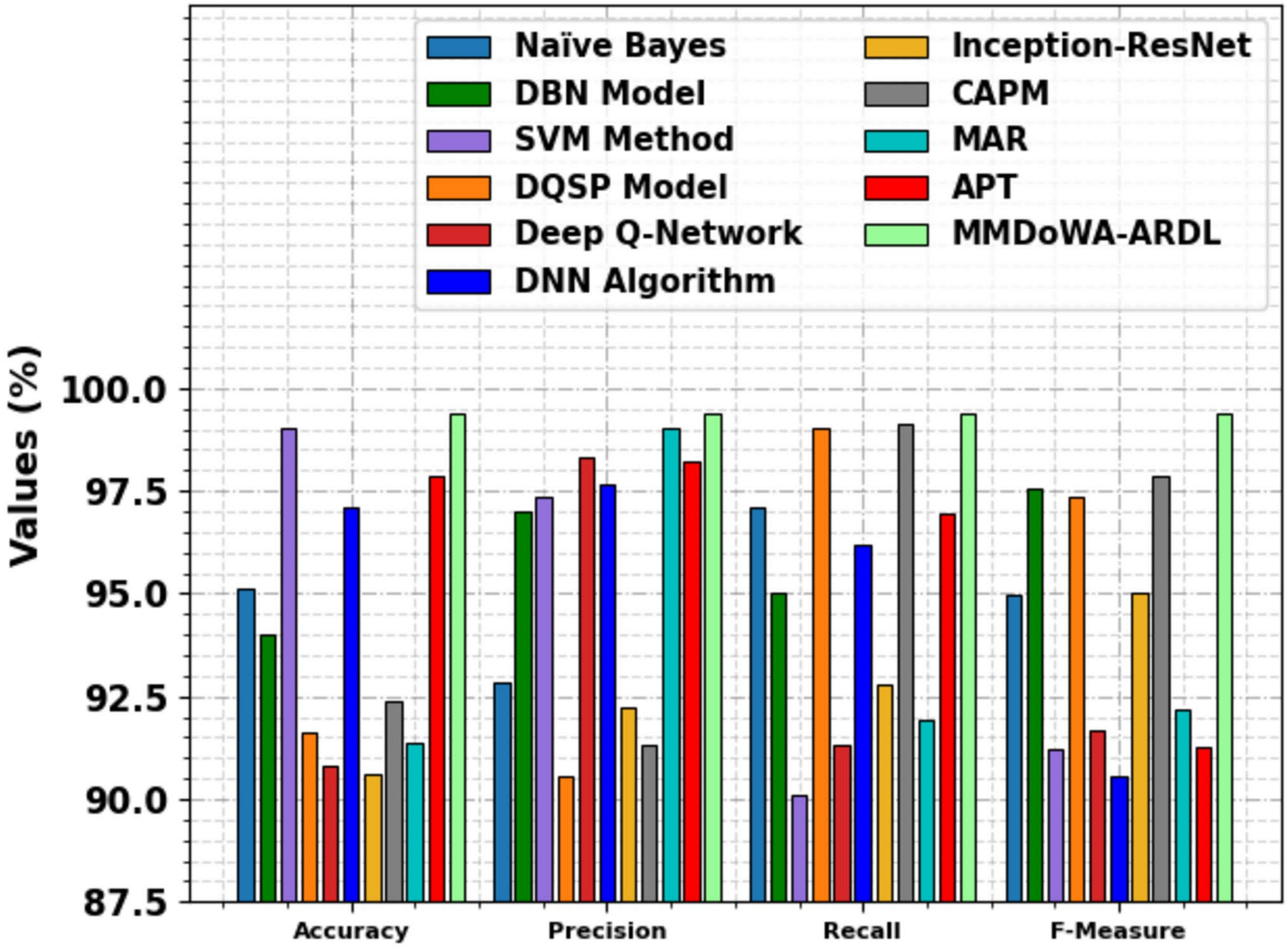


Fig. 11. Comparative outcome of MMDoWA-ARDL approach with existing methods.

Classifier	CT (sec)
NB	8.60
DBN Model	9.63
SVM Method	8.74
DQSP Model	19.52
Deep Q-Network	17.37
DNN Algorithm	16.68
Inception-ResNet	18.32
CAPM	11.90
MAR	10.56
APT	12.25
MMDoWA-ARDL	5.72

Table 7. CT outcome of MMDoWA-ARDL technique with existing models.

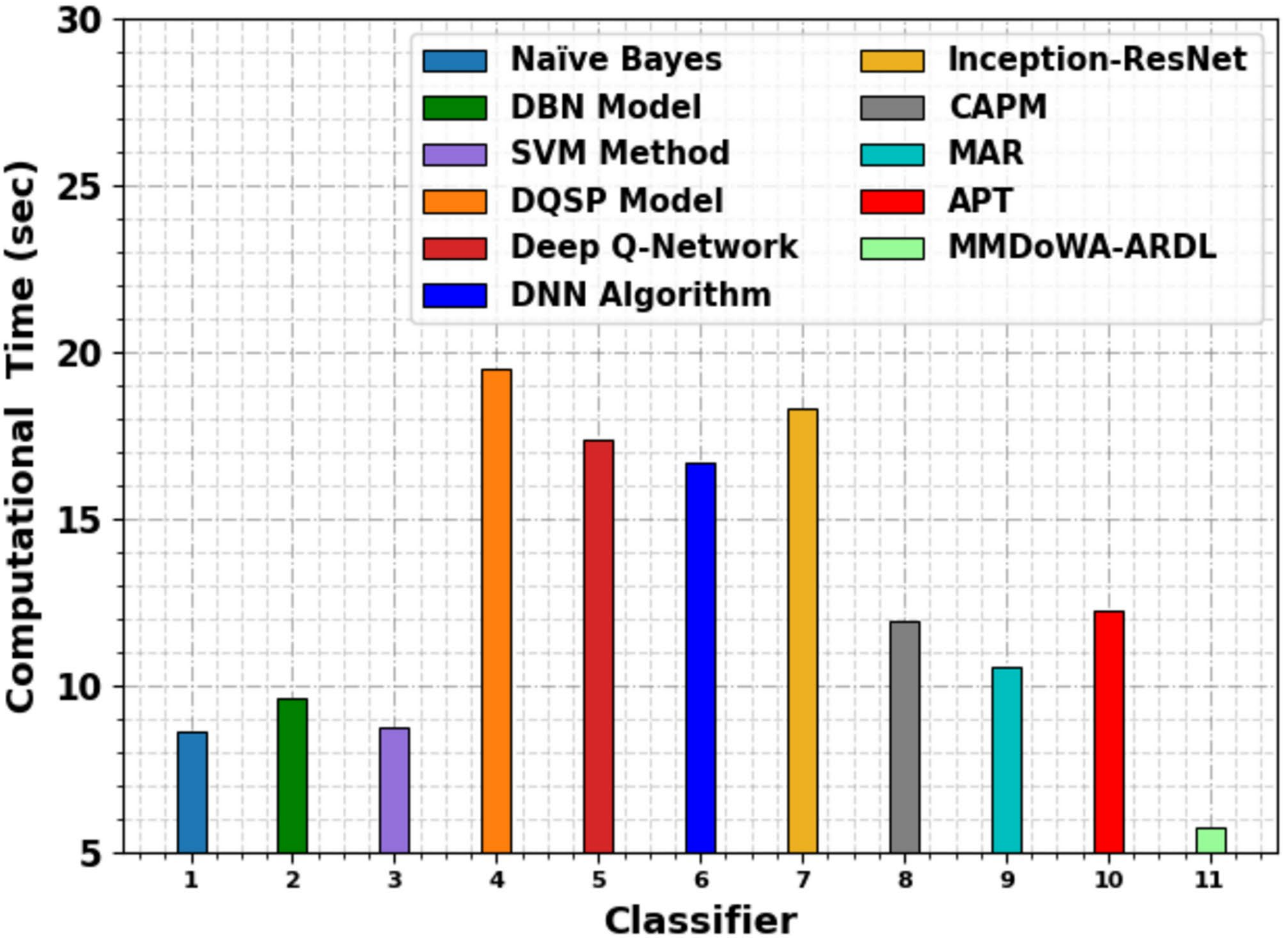


Fig. 12. CT outcome of MMDoWA-ARDL technique with existing models.

Classifier	$Accu_y$	$Prec_n$	$Reca_l$	$F_{Measure}$
CSO	97.35	97.33	97.12	97.48
SBOA	97.86	98.05	97.88	98.03
BMNet	98.65	98.84	98.62	98.6
MMDoWA-ARDL	99.39	99.39	99.39	99.39

Table 8. Result analysis of the ablation study of MMDoWA-ARDL approach.

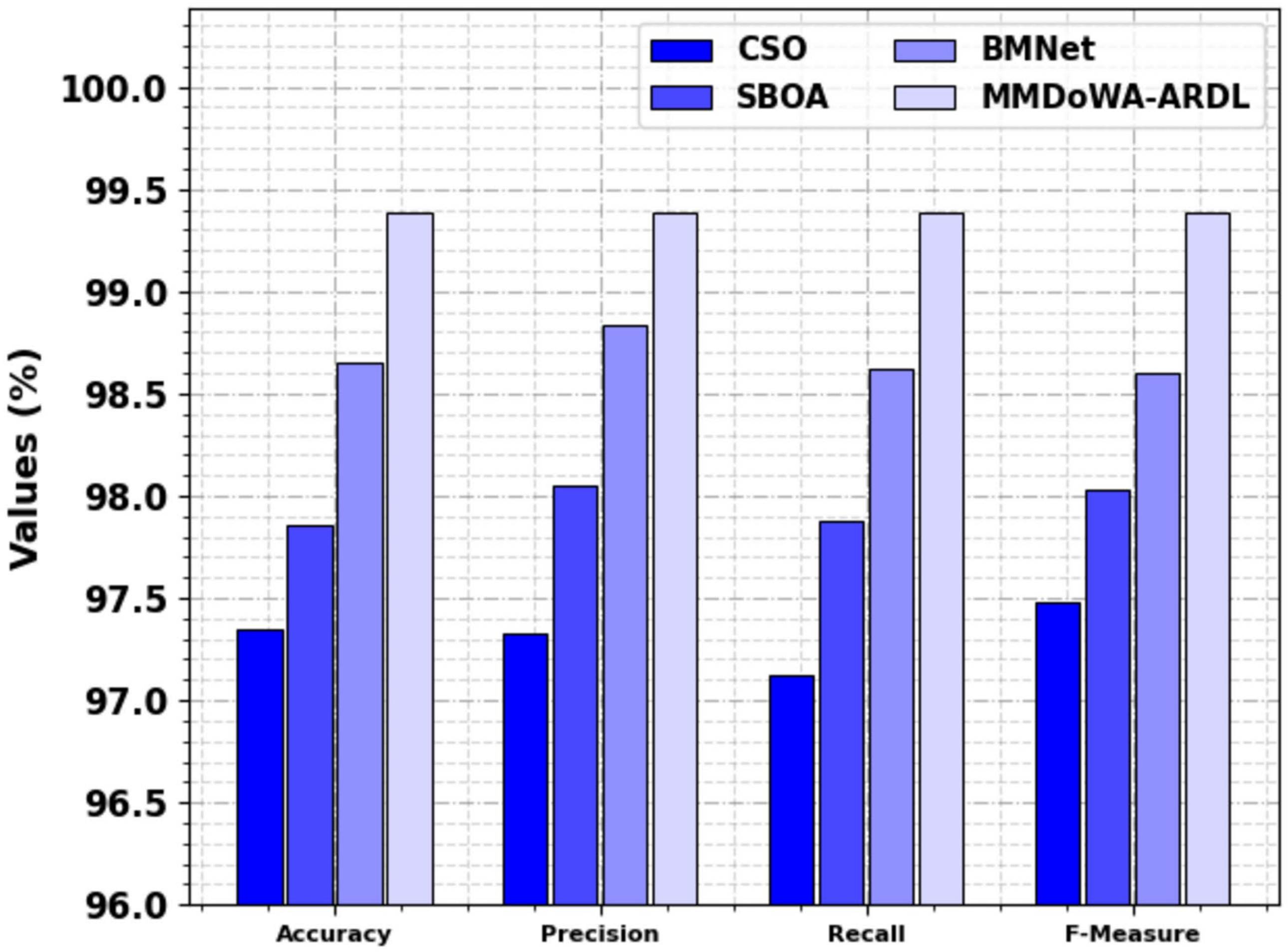


Fig. 13. Result analysis of the ablation study of MMDoWA-ARDL approach.

Data availability

The data supporting this study's findings are openly available at https://data.mendeley.com/datasets/g8g9vdxyvn/1, reference number [33].

Received: 13 March 2025; Accepted: 5 May 2025
Published online: 28 May 2025

References

1. Kelly, D., Barrett, E. & Glavin, F. *Denial of Wallet: Analysis of a Looming Threat* (and Novel Solution for Mitigation using Image Classification, 2023).
2. Raikwar, M. & Gligoroski, D. August. DoS attacks on blockchain ecosystem. In *European Conference on Parallel Processing* (pp. 230–242). Cham: Springer International Publishing. (2021).
3. Chaganti, R. et al. A comprehensive review of the denial of service attacks in blockchain ecosystem and open challenges. *IEEE Access.* **10**, 96538–96555 (2022).
4. Shen, J. et al. November. Gringotts: fast and accurate internal denial-of-wallet detection for serverless computing. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security* (pp. 2627–2641). (2022).
5. Baek, U. J. et al. September. DDoS attack detection on the Bitcoin ecosystem using deep learning. In *2019 20th Asia-Pacific network operations and management symposium (apnoms)* (pp. 1–4). IEEE. (2019).
6. Bhardwaj, A., Mangat, V., Vig, R., Halder, S. & Conti, M. Distributed denial of service attacks in the cloud: State-of-the-art of scientific and commercial solutions. *Computer Science Review* **39**, 100332. (2021).
7. Mileski, D. & Mihajloska, H. November. Distributed denial of wallet attack on serverless pay-as-you-go model. In *2022 30th Telecommunications Forum (TELFOR)* (pp. 1–4). IEEE. (2022).
8. Kelly, D., Glavin, F. G. & Barrett, E. DoWTS–Denial-of-Wallet test simulator: Synthetic data generation for preemptive defence. *J. Intell. Inform. Syst.* **60**(2), 325–348 (2023).
9. Musa Baig, S., Javed, M. U., Almogren, A., Javaid, N. & Jamil, M. A blockchain and stacked machine learning approach for malicious nodes' detection in internet of things. *Peer-to-Peer Netw. Appl.* **16**(6), 2811–2832 (2023).
10. Sait, A. R. W., Pustokhina, I. & Ilayaraja, M. Mitigating DDoS attacks in wireless sensor networks using heuristic feature selection with deep learning model. *Full-Length Article* **2**, 65 – 5 (2021).
11. Lu, T. et al. Intelligent upgrading of the localized GNSS monitoring system: Profound integration of blockchain and AI. *Electronics* **14**(3), 490 (2025).

12. Aguru, A. & Erukala, S. OTI-IoT: A blockchain-based operational threat intelligence framework for multi-vector DDoS attacks. *ACM Trans. Internet Technol.* (2024).
13. Ullah, F. et al. Ehr management evolution through purpose-based access control and blockchain smart contracts. *Int. J. Inf. Secur.* **24**(1), 63. (2025).
14. de Lima, A. C., Alchieri, E. A., Bordim, J. L. & Gondim, J. J. November. An improved method for detecting EDoS attacks in the cloud with hyperparameter optimization and metaheuristic algorithms. In *2024 Twelfth International Symposium on Computing and Networking Workshops (CANDARW)* (pp. 43–49). IEEE. (2024).
15. Manh, B. D. et al. Privacy-Preserving cyberattack detection in Blockchain-Based IoT systems using AI and homomorphic encryption. *IEEE Internet Things J.* (2025).
16. Sangore, R. B. & Patil, M. E. October. Boosting Bitcoin Security: Innovative Threat Detection with Ensemble Learning and Water Cycle Algorithm Optimization. In *2024 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS)* (pp. 1–6). IEEE. (2024).
17. Almazroi, A. A. Innovative AI ensemble model for robust and optimized blockchain-based healthcare systems. *Netw. Model. Anal. Health Inf. Bioinf.* **14**(1), 1–19 (2025).
18. Rani, U. et al. An optimized neural network with adahessian for cryptojacking attack prediction for securing crypto exchange operations of MEC applications. *Journal of Cloud Computing* **13**(1), 63 (2024).
19. Renukadevi, P., Amaran, S., Vikram, A., Rao, T. P. & Ishak, M. K. Enhancing cybersecurity through fusion of optimization with deep wavelet neural networks on denial of wallet attack detection in serverless computing. *IEEE Access.* **13**, 47111–47122 (2025).
20. Ramos, S., Pianese, F., Leach, T. & Oliveras, E. A great disturbance in the crypto: Understanding cryptocurrency returns under attacks. *Blockchain: Research and Applications* **2**(3), 100021. (2021).
21. Li, C. et al. FaaSMT: Lightweight serverless framework for intrusion detection using merkle tree and task inlining. *arXiv preprint arXiv:2503.06532*. (2025).
22. Joshi, C. et al. October. High-fidelity simulated dataset for enhanced detection of denial of wallet attacks (DOW) in serverless architecture. In *2024 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS)* (pp. 1–6). IEEE. (2024).
23. Krishna, E. P., Sandhya, E. & Priya, K. L. Cutting-Edge approaches to data protection and encryption in cloud computing security. In Convergence of Cybersecurity and Cloud Computing (303–324). IGI Global Scientific Publishing. (2025).
24. Mora, H., Ortega Candel, J. M., Mora Gimeno, F. J., Maciá, A. & Abaldea, A. V., Dataset para análisis y detección de amenazas Denial of Wallet (DOW) en arquitecturas Serverless Computing. (2024).
25. Ranganatha, H. R. & Mustafa, A. S. Enhancing fraud detection efficiency in mobile transactions through the integration of bidirectional 3d Quasi-Recurrent Neural network and blockchain technologies. *Expert Systems with Applications*, *260*, 125179. (2025).
26. Xiong, J., Wei, M., Lu, Z. & Liu, Y. *Warmonger Attack: A Novel Attack Vector in Serverless Computing* (IEEE/ACM Transactions on Networking, 2024).
27. Musamih, A. et al. Blockchain and NFT-based solution for genomic data management, sharing, and monetization. *IEEE Access.* (2025).
28. Lavi, D., Brodt, O., Mimran, D., Elovici, Y. & Shabtai, A. Detection of Compromised Functions in a Serverless Cloud Environment. *arXiv preprint arXiv:2408.02641*. (2024).
29. Alrslani, F. A. et al. Enhancing cybersecurity via attribute reduction with deep learning model for false data injection attack recognition. *Sci. Rep.* **15**(1), 3944 (2025).
30. Keerthi Guttikonda, D.R.G. & Prasad, D.G. Cuckoo search optimization-based feature selection for predicting autism spectrum disorder using artificial immune algorithms. *J. Theoretical Appl. Inform. Technol.* **103**(2) (2025).
31. Xiong, D., Wen, Z., Zhang, C., Ren, D. & Li, W. BMNet: Enhancing deepfake detection through BiLSTM and multi-head self-attention mechanism. *IEEE Access.* (2025).
32. Zhu, Y. et al. Secretary bird optimization algorithm based on quantum computing and multiple strategies improvement for KELM diabetes classification. *Sci. Rep.s 15*(1), 3774 (2025).
33. https://data.mendeley.com/datasets/g8g9vdxyvn/1
34. Candel, J. M. O., Gimeno, F. J. M. & Mora, H. M. Generation of a dataset for DoW attack detection in serverless architectures. *Data in Brief*, *52*, p.109921. (2024).
35. Elsadig, M. A. Detection of Denial-of-Service attack in wireless sensor networks: A lightweight machine learning approach. *IEEE Access.* (2023).
36. Jang, Y. et al. An investigation of learning model technologies for network traffic classification design in cyber security exercises (IEEE Access, 2023).
37. Shitharth, S. et al. An enriched RPCO-BCNN mechanisms for attack detection and classification in SCADA systems. *IEEE Access.* **9**, 156297–156312 (2021).

Acknowledgements

The authors extend their appreciation to the Deanship of Research and Graduate Studies at King Khalid University for funding this work through Large Research Project under grant number RGP2/51/45. Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2025R721), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. Researchers Supporting Project number (RSP2025R459), King Saud University, Riyadh, Saudi Arabia. The authors extend their appreciation to the Deanship of Scientific Research at Northern Border University, Arar, KSA for funding this research work through the project number NBU-FFR-2025-1661-04. The authors are thankful to the Deanship of Graduate Studies and Scientific Research at University of Bisha for supporting this work through the Fast-Track Research Support Program.

Author contributions

Amal K. Alkhalifa: Conceptualization, methodology development, experiment, formal analysis, investigation, writing. Mohammed Aljebreen: Formal analysis, investigation, validation, visualization, writing. Nazir Ahmad: Formal analysis, review and editing. Sultan Alahmari: Methodology, investigation. Othman Alrusaini: Review and editing.Ali Alqazzaz: Discussion, review and editing. Hassan Alkhiri: Discussion, review and editing. Rakan Alanazi: Conceptualization, methodology development, investigation, supervision, review and editing.All authors have read and agreed to the published version of the manuscript.

Declarations

Competing interests

The authors declare no competing interests.

Ethics approval

This article does not contain any studies with human participants performed by any of the authors.

Additional information

Correspondence and requests for materials should be addressed to R.A.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.