Metaheuristic Optimization Algorithms for Deep Learning Model Design in Secure Internet of Things Environment

Mustafa Hussein Zwayyer, Sajida Allawi Dawood and Ammar Bassem Saleh
College of Islamic Science, University of Baghdad, 10071 Baghdad, Iraq
mustafa.h@cois.uobaghdad.edu.iq, Sajida.a@cois.uobaghdad.edu.iq, amar.saleh@cois.uobaghdad.edu.iq

Keywords: Botnet Detection, Internet of Things (IoT), Metaheuristic Optimization, Deep Learning, Feature Selection.

Abstract: The Internet of Things (IoT) has enabled smart systems, but it has also increased vulnerabilities to cyber threats, including botnet attacks. To address these security challenges, this study proposes a hybrid system that combines metaheuristic and machine learning. To tune hyperparameters of a hybrid neural network based on Convolutional Neural Networks and Semi-Recurrent Neural Networks (CNN-QRNN), the Chaotic Butterfly Optimization Algorithm (CBOA) is used. A new metaheuristic algorithm, Self-Adaptive Enhanced Harris Hawks Optimization (SAEHO), as well as a self-upgraded cat and mouse optimizer (SU-CMO), are introduced and evaluated in order to enhance model effectiveness. Based on experiments conducted on the N-BaIoT dataset, it was determined that the proposed models significantly outperformed conventional classifiers in key performance metrics, including accuracy, the Matthews Correlation Coefficient (MCC), the Rand Index, and the F-Measure. Particularly notable improvements were observed in reducing false-positive rates and enhancing anomaly detection sensitivity. The HMMLB-BND method substantially improves detection performance in diverse IoT environments, offering a robust, efficient, and scalable solution suitable for real-time deployment in resource-constrained systems.

1 INTRODUCTION

Through IoT, devices and systems can be connected, data can be processed in real time, and intelligent decisions can be made. With the rise of interconnected devices, security against a wide range of cyber threats becomes increasingly important. For IoT environments, optimizing deep learning models poses a crucial challenge, especially in terms of security and performance. A powerful tool for addressing this challenge has emerged: metaheuristic optimization algorithms. These algorithms provide solutions to complex optimization problems that traditional methods may not be able to solve, such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Simulated Annealing (SA) [1]. As a result of their adaptability, robustness, and ability to explore large solution spaces, they are ideal candidates for enhancing deep learning models in IoT applications. Optimization techniques are essential for fine-tuning hyperparameters, network architectures, and training strategies in deep learning models to ensure optimal performance and security. With IoT algorithms, deep learning models are more accurate and efficient and better defended against threats, ensuring privacy and data integrity.

Recently, researchers have become increasingly interested in cloud computing (CC) [2]. With the ability to provide on-demand resources and services, the ability to operate data centres, power systems, intelligent transportation systems, video delivery systems, earthquake command systems, and other technological environments has been impacted [3]. A CC model considers a number of design objectives, including energy efficiency, fairness, reliability, and fault tolerance. Security, however, is considered the most important design objective. It is common for IoT to appear in CC ecosystems [4]. The technology aggregates geographically dispersed cyber-enabled systems or cyber-physical devices to provide strategic services. A major goal of the IoT platform is security, as with CC [5]. Cyberattacks are becoming increasingly prevalent on IoT devices [6], [7]. The internet is interconnected with many IoT gadgets, making it easier to abuse security controls [8]. IoT poses several security threats, including a number of vulnerabilities. The IoT is prone to a variety of attacks, so categorizing them and identifying appropriate vulnerabilities is important. IoT systems are likely to be affected by routing, jamming, sinkholes, DoS, wormholes, worm attacks, flooding, and viruses, according to some research [9]. It is more specific to IoT platforms in production that are attacked and flooded with DoS attacks [10], [11].

In recent years, botnet attacks have become increasingly popular. The botnet damages services by causing disruptions and depleting resources. Such IoT attacks are commonly discovered using artificial intelligence. Through intrusion detection systems (IDS), malicious activity can be detected on networks. According to the detection system, IDS fall into two categories: anomaly-based and signature-based [12], [13]. As a first step, we can use baseline network behaviour. In this method, malicious events, both unknown and known, will be detected. In the next step, a specific pattern from the network (such as a sequence of bytes) will be applied. Afterwards, it compares the sequences to current databases of signatures [14]. A recent study found that deep learning (DL) is a better method for detecting IoT assaults than traditional machine learning (ML). However, the cloud layer can only run these algorithms because of its limited resources. Moreover, in some cases, such as remote live functions, these methods do not work because the system is assumed to make realistic decisions faster.

As IoT devices and cloud services converge, the ecosystem faces several complexities due to its scale, heterogeneity, and dynamic nature. It is difficult to devise universal detection methods for IoT devices because of the sheer number of different communication protocols and abilities. In addition to encrypted communication between devices and cloud services, it is also difficult to detect botnet activity in network traffic. As IoT devices have limited resources, it isn't easy to design resource-intensive detection approaches, so lightweight yet effective approaches need to be designed. Since botnets are distributed and can mimic legitimate device behaviour, it's difficult to identify malicious activity and command-and-control nodes. IoT devices lack uniform security standards, which exacerbates the detection of botnets due to the rapid development of attack approaches [15].

This issue is addressed by designing a hybrid metaheuristic based on machine learning called HMMLB-BND that is coupled with the Cloud Assisted Internet of Things (CASIoT). The present HMMLB-BND approach enables feature selection (FS) by using the modified firefly optimization (MFFO) technique [16]. In the HMMLB-BND technique, convolutional neural networks (CNNs) and quasi-recurrent neural networks (QRNNs) are used to detect botnets based on a hybrid algorithm that combines these two types of networks. Hyperparameter optimization uses CBOA. Simulating the N-BaIoT data demonstrated the enhanced performance of HMMLBBND. Here are the key contributions:

- The purpose of this paper is to present a new HMMLB-BND technique that is based on MFFOs that select features for botnet detection, CNNs-QRNNs that classify users, and CBOAs that tune hyperparameters for botnet detection. We are unaware of any literature descriptions of HMMLB-BND.
- MFFO algorithm is presented for feature selection, which overcomes the issue of ineffective exploration capability and local optima.

2 LITERATURE REVIEW

They propose a whale optimization algorithm (WOA) that uses nature-inspired meta-heuristics [17]. A whale-inspired algorithm solves the target problem by imitating whales' predatory behaviour. Using bubble nets and two operators, the device mimics the behaviour of humpback whales, including searching for prey and surrounding prey. Thirty-nine problems are evaluated as part of the WOA evaluation, 29 of which involve mathematical optimization challenges, and 6 which involve structural design challenges. WOA has the potential to solve problems involving unknown search spaces, as shown by the results of the study. In WOA, there is a slow convergence.

Based on the behaviour of grey wolves, the author [18] developed a metaheuristic optimization algorithm known as grey wolf optimization (GWO). The simulation is based on real-life social structures and grey wolves' hunting processes. To mimic the hierarchy of leadership, grey wolves are divided into four types: beta, delta, and omega. Mathematically, alpha wolves are the most optimal solution. According to experts, beta wolves, delta wolves, and omega wolves are the second, third, and fourth best solutions. Despite its popularity, however, this algorithm suffers from low precision, slow convergence, and a poor local search capability.

Based on the Salp Swarm Algorithm (SSA), a modern metaheuristic algorithm has been developed for simulating salps in deep water [19]. In several FS algorithms, SSA is used as a search strategy [20]. According to [21], who addressed the problems of the SSA algorithm, this trend of opportunistic search behaviour improvement was also observed. Local search (LS) improved the SSA's ability to be exploited. In addition, a chaotic map and a new equation variable were used to find the most effective location update method. To evaluate the effectiveness of the recommended technique for feature selection, a benchmark categorization dataset and three Hadith datasets were analyzed. A dynamic SSA performed better than alternative solutions.

Using Restricted Boltzmann Machines (RBMs), an SC intrusion detection system was proposed in [22]. With RBMs, high-level features can be learned from raw data, and real-time data representations can be controlled. The authors [23] investigate ML-based techniques for detecting hijackings, GPS jamming, and denial-of-service attacks that can be used against drone attacks. Classification of DJI Phantom 4 methods is based on machine learning classifiers that can work with either normal or malicious signatures [24]. Using a three-stage approach to investigating, reducing, and classifying data traffic [25], intrusion detection is achieved by distinguishing positive trust service requests from false ones. To reduce data and drive classifiers accordingly, the solution utilizes decision trees and deep belief algorithms from machine learning. Through simulations, solutions for detecting intrusion attacks are validated to demonstrate their efficiency.

To solve this problem, the author [26] proposed a hybrid method and an infrastructure method. First, the BoT-IoT identify database was executed, and its 44 effective attributes were included in the ML technique. Five effective machine-learning techniques and widely used performance evaluation metrics were selected to identify malicious and anomalous traffic. Using a smart city platform that detects DDoS and replay attacks in real time, a hybrid DL technique was formulated [27]. SC data (smart soil, smart environment, and Smart River) are available for testing the hybrid method; replay attacks and DDoS attacks can be simulated. The anchor node is a crucial part of the sensor network's cost-effectiveness and energy efficiency because it influences localization in a significant way.

Assaults in today's society can be classified using DL-related techniques based on recent databases [28]. A safeguard was introduced to protect the IoT network's reputation and ensure its access is restricted to those who need it. The base was presented for incorporating IDS into IoT-related networks [29]. In [30], the authors propose a mechanism for detecting botnets at the application layer of DNS services by using a two-level DL structure. The similarity of DNS requests across Ethernet connections was determined using a Siamese network based on an existing threshold. On the second level of the structure, domain generation is recommended as an alternative to DL architectures to categorize abnormal and normal domain names. By using computing to address some of the current limitations and issues with IoT solutions for smarter cities, we can enhance them and make them more useful. Utilizing the Internet of Things and cloud computing, smart cities will have the ability to deliver novel and enhanced services by leveraging big data stored in the cloud.

3 METHODOLOGY

3.1 An Algorithm for Optimizing the Herding of Elephants Based on the Seagull Adaption (SAEHO)

Based on the combination of Elephant Herding Optimization (EHO) and Self-Adaptive Optimization (SAO), the hybrid SAEHO model was developed. EHO cannot make use of real-time or predictive data to determine ongoing and future search directions. On the other hand, SAO may lead to poor performance in complex optimization tasks due to slow computation times and cumbersome constraints. Taking advantage of both methods, the SAEHO model addresses these challenges. Hybrid models, such as SAEHO, have demonstrated promise for improving optimization and search outcomes. Elephant herds exhibit a harmonious interaction with their environment and engage in clan leadership led by a female matriarch in EHO. As a reflection of this social structure, three basic behavioural rules guide the EHO algorithm.

The motivation and mathematical modelling for the proposed algorithm are discussed in this section.

There are many different species of seagulls around the world, all belonging to the Laridae family of birds. It is possible to find a wide range of seagull species with different masses and lengths. Besides eating insects and fish, seagulls also eat reptiles, amphibians, earthworms, and other species of animals. A seagull is a very intelligent bird. To attract earthworms, they produce a rain-like sound with their feet and use breadcrumbs to attract fish. Neither fresh nor salt water is off-limits to seagulls. The majority of animals are incapable of doing this. The bill of seagulls is equipped with a series of glands that flush salt from the system through openings above their eyes.

3.2 Mathematical Model

There is a discussion of the mathematical models of migration and attack on the prey:

- Migration (exploration). This algorithm simulates how the group of seagulls moves from place to place during migration. There are three conditions a seagull must meet during this phase:
- Avoiding the collisions. A variable A is incorporated into the calculation of the new position of the search agent to prevent collisions with neighbouring seagulls (i.e., other seagulls).

  $$C_s = A * P_s(x). \tag{1}$$

  The (1) expresses C_s as the position of search agent, which avoids colliding with other search agents, P_s as its current position, x as its current iteration, and A as its movement strategy within a search space.

- The movement towards the best neighbour's direction. When the search agents have avoided colliding with neighbours, they move toward the direction of the nearest neighbour.

  $$M_s = b * \left(p_{bs}(x) - P_s(x)\right), \tag{2}$$

  Search agent P_s is positioned toward the best-fitting search agent (e.g., the fittest seagull) by M_s.

Randomness is responsible for the proper balance between exploration and exploitation. B is calculated as:

$$B = 2 * A^2 * rd. \tag{3}$$

A matriarch leads each elephant clan in accordance with elephant natural habits. Matriarch ci influences each elephant's new position, therefore. Ib (4) calculates elephant j in clan ci.

$$x_{new,ci.f} = x_{ci,j} + a * \left(x_{best,ci} - x_{ci,j}\right) * r. \tag{4}$$

The new position and old position of elephant j in clan ci, are represented by $x_{new,ci.f}$and $x_{ci,j}$, respectively. Elephant x_{best,ci^I} am the matriarch of the clan, representing the strongest elephant. An indication of scale factor, $r \in [0,1]$, can be seen as $a \in [0,1]$. For each clan, (5) can be used to determine which elephant is the best.

$$x_{new,ci,j} = \beta * x_{center,ci}\,. \tag{5}$$

Where $\beta \in [0,1]$ is a factor that determines how $x_{center,ci}$ affects $x_{new,ci,j}$. An individual named $x_{new,ci,j}$ has been born. Clan ci is centred around. $x_{center,ci}$. The (6) can be used to calculate it.

$$x_{center,ci,d} = \frac{1}{n_{ci}} * \sum_{j=1}^{n_d} x_{ci,j,d}\,. \tag{6}$$

The number of elephants per clan is given by $1 \leq d \leq D$ and n_{ci}. Individual elephants are characterized by their d^{th} dimension $x_{ci,j,d}$. The (6) can be used to update $x_{center,ci}$, which is the clan centre.

Solving optimization problems, one can model the separating process of male elephants leaving their families. Elephants with the worst fitness during each generation implement the separating operator, as shown in (7).

$$x_{worst,ci} = x_{min} + (x_{max} - x_{min} + 1) * rand. \tag{7}$$

3.3 Self-Upgraded Cat and Mouse Optimizer (SU-CMO) Algorithm

However, the current CMBO model tolerates low accuracy while offering optimal solutions. A number of improvements were made to conventional CMBO in order to overcome its drawbacks. As shown in the SU-CMO flowchart and model, the following steps are involved.

We are in the process of initializing the population of BB search agents. Initialization of B, B_c, B_m, T parameters. B indicates how many members are in population matrix A.

According to (8), the initial population is created.

$$A = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_B \end{bmatrix}_{B*m} = \begin{array}{ccccc} y_{1,1} & \cdots & y_{1,d} & \cdots & y_{1,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ y_{i,1} & \cdots & y_{i,d} & \cdots & y_{i,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \lfloor y_{B,1} & \cdots & y_{B,d} & \cdots & y_{B,m} \rfloor \end{array}_{B*m} \tag{8}$$

A variable named $y_{i,d}$ is used here to represent the d^{th} problem.

Based on (9), the fitness of search agents is computed.

$$Obj = \min(Err). \tag{9}$$

The sorted population matrix A^S should be updated using (10) and (11). As shown here, as $y^S_{1,d}$ represents the sorted population matrix for the populationi^{th}.

Moreover, Ob^s_j represents the vector of objective functions sorted by

$$A^S = \left\lfloor \begin{matrix} A_1^S \\ A_2^S \\ \vdots \\ A_B^S \end{matrix} \right\rfloor_{B*m} = \left\lfloor \begin{matrix} y_{1,1}^S & \cdots & y_{1,d}^S & \cdots & y_{1,m}^S \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ y_{i,1}^S & \cdots & y_{i,d}^S & \cdots & y_{i,m}^S \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ y_{B,1}^S & \cdots & y_{B,d}^S & \cdots & y_{B,m}^S \end{matrix} \right\rfloor_{B*m}, \quad (10)$$

$$Obj^S = \left\lfloor \begin{matrix} Obj_1^S & \min(Obj) \\ Obj_2^S & \min(Obj) \\ \vdots & \vdots \\ Obj_B^S & \min(Obj) \end{matrix} \right\rfloor_{B*1}. \quad (11)$$

A mouse population is selected based on (12).

$$C = \left\lfloor \begin{matrix} C_1 = X_{Bm+1}^S \\ \vdots \\ C_i = X_{Bm+j}^S \\ \vdots \\ C_{B_c} = X_{Bm+B_c}^S \end{matrix} \right\rfloor_{B_c+m} = \quad (12)$$

$$= \left\lfloor \begin{matrix} y_{Bm+1,1}^S & \cdots & y_{Bm+1,d}^S & \cdots & y_{Bm+1,m}^S \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ y_{Bm+j,1}^S & \cdots & y_{Bm+j,d}^S & \cdots & y_{Bm+j,m}^S \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ y_{Bm+B_c,1}^S & \cdots & y_{Bm+B_c,d}^S & \cdots & y_{Bm+b_c,m}^S \end{matrix} \right\rfloor_{B_c*m}$$

Accordingly, M, B_m, M_i, C, B_c, C_j indicates the number of mice, the count of mice, the j^{th} mice, the number of cats, and the i^{th} cat. The (13) shows how the positions of cats are updated, where C_j^{new} new points to the j^{th} cat's new position, and $C_{j,h}$ is the d^{th} cat's new value. Also, r is estimated at random within the range [0, 1]. An integer is used as a random variable in (14).

$$C_j^{new} = \left[C_{j,d} + r * \left(M_{k,d} - I * C_{j,d}\right)\right]. \quad (13)$$

Here,

$$I = round(1 + rand), \quad (14)$$

If $j = B_c$

Assuming the above conditions are met, (15) is used to create H_i.

$$H_i = h_{i,d} = y_{i,d} \& i = 1{:}\, B_{m,} d = 1{:}\, m, l{:} \in 1{:}\, B. \quad (15)$$

Following that, mouse positions are updated using (16) and (17). The (17) shows the typical method for updating M_i. ra_1 and ra_2 are random integers used in the SU-CMO model to update M_i. It is assumed that ra_1 and ra_2 have values of 1.25 and 1.75 respectively

$$M_i^{new}{:}\, m_{i,d}^{new} = m_{i,d} + r * \left(h_{i,d} - I * m_{i,d}\right) + Sign(F_i^m - F_i^H), \quad (16)$$

$$M_i = \begin{cases} M_i^{new} | F_i^{m,new} & < F_i^m \\ M_i & |else \end{cases},$$

$$M_i = \begin{cases} M_i^{new} | F_i^{m,new} & .ra_1 < F_i^m \\ M_i & |else \end{cases}, \quad (17)$$

$$\mathrm{M_i} = \begin{cases} \mathrm{M_i^{new} | F_i^{m,new}} & \mathrm{.ra_1 < F_i^m} \\ \mathrm{M_i} & \mathrm{|else} \end{cases},$$

$$M_i = \begin{cases} M_i^{new} | F_i^{m,new} & .ra_2 < F_i^m \\ M_i & |else \end{cases}.$$

4 RESULTS AND DISCUSSION

IoT attacks are organized into eight specific categories: Analysis, Backdoors, Denial of Service, Exploits, Fuzzers, and Reconnaissance. Although they are divided into distinct categories, they are all grouped under the overarching category of "Attack." The chart in Figure 1 shows how these attack types are categorized in Datasets 1 and 2.

In Dataset 1, hybrid classifiers with 80% learning rates were evaluated using the SAEHO algorithm. DTs, SVMs, NNs, SLnOs, and Grey Wolf Optimizers are included in the comparison in addition to Proposed Model 1. The two performance metrics used for comparison are accuracy and Matthews Correlation Coefficient (MCC), as shown in Figure 2.. A consistent winner among all the classifiers is the Proposed Model 1 across all metrics. This method achieves the highest levels of Accuracy and Rand Index, indicating that it is reliable for classification and clustering. It also ranks first in F-Measure, indicating a good balance between precision and recall, as well as a significantly higher MCC. According to these results, Proposed Model 1 is superior to the given experimental setup in detecting IoT intrusions.

In Figure 3, the 80% learning rate SAEHO algorithm is used to perform a performance comparison of hybrid classifiers on dataset 2. There are six classifiers evaluated across four metrics: In addition to accuracy, MCC, Rand Index, and F-measure, proposed models include DT, SVM, NN, SLnO, GWO, and WOA. The Rand Index, MCC, and F-Measure consistently outperform the other methods, with near-perfect accuracy in each. As a result of these results, it is demonstrated that the system is robust and effective in detecting IoT attacks.

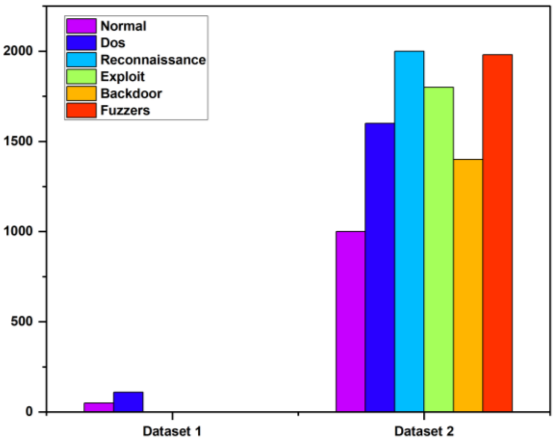

Figure 1: A number of attacks were recorded in Datasets 1 and 2.

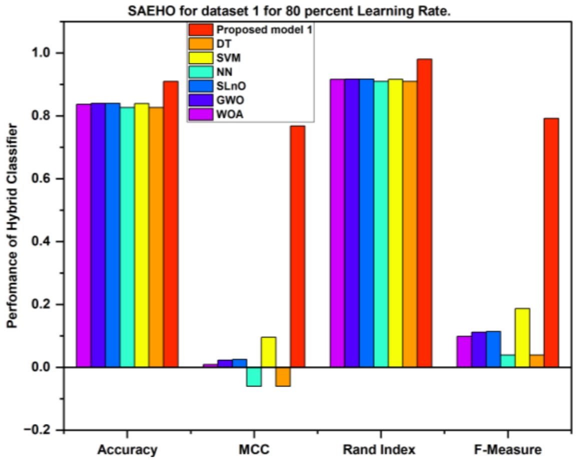

Figure 2: Performance comparison of hybrid classifiers on dataset 1 with 80% learning rate.

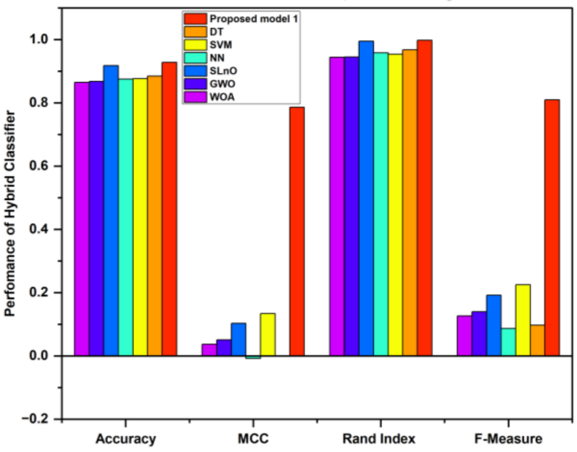

Figure 3: Performance comparison of hybrid classifiers with 70% learning rate.

A hybrid classifier using the SUC-MO method with an 80% learning rate is shown in Figure 4. ALO, GMBO, RNN, NN, and AO were evaluated on Accuracy, MCC, Rand Index, and F-Measure, in addition to the Proposed Model 2. It achieves the highest F-Measure and strong Accuracy and MCC results in comparison with the other models. All of the metrics indicate that Proposed Model 2 performs the most effectively compared to GMBO despite GMBO having a marginally higher Rand Index.

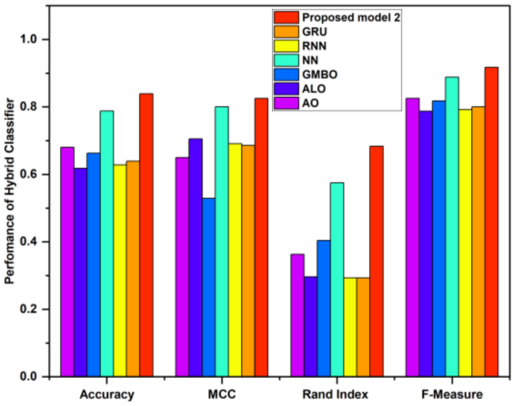

Figure 4: Performance comparison of hybrid classifiers with 75% learning rate.

Dataset 2 is shown in Figure 5 with hybrid classifiers using SUC-MO algorithms at an 80% learning rate. Comparisons have been made between six different models: Proposed Model 2, RNN, NN, GMBO, ALO, and AO in terms of accuracy, as well as MCC, Rand Index, and F-Measure in terms of reliability. Model 2 outperforms others overall, achieving the highest F-Measure and MCC, as well as high accuracy and Rand Index, which demonstrate its effectiveness for classifying IoT data

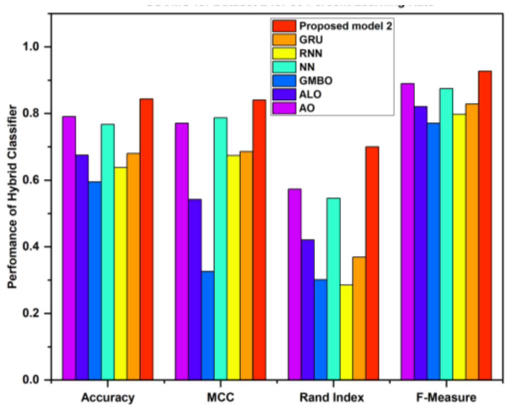

Figure 5: Performance comparison of hybrid classifiers with 80% learning rate.

5 CONCLUSIONS

An advanced metaheuristic optimization technique and deep learning models are combined to provide comprehensive botnet detection for cloud-based IoT systems. In the proposed HMMLB-BND method, MFFO is utilized for optimizing feature selection, enhancing the relevance and reduction of redundant information. CNN-QRNN is employed to achieve accurate and reliable classification by effectively capturing spatial and sequential dependencies within the data. Additionally, the CBOA algorithm is integrated for precise tuning of hyperparameters, ensuring optimal model performance. Furthermore, the study explores advanced metaheuristic algorithms such as SAEHO and SU-CMO to further enhance detection accuracy, robustness, and adaptability to varying IoT scenarios. Comprehensive experimental evaluations demonstrate that the proposed models significantly outperform traditional and hybrid classifiers when tested on the N-BaIoT dataset, exhibiting substantial improvements particularly in terms of Matthews Correlation Coefficient (MCC) and F-Measure. Due to its capability to effectively address critical challenges such as high-dimensional datasets, imbalanced classification, limited computational resources, and stringent real-time requirements inherent in IoT environments, the HMMLB-BND framework presents a highly promising direction for future research endeavors and practical real-time deployment within the evolving field of IoT security.

REFERENCES

[1] P. Rani, P. N. Singh, S. Verma, N. Ali, P. K. Shukla, and M. Alhassan, “An implementation of modified blowfish technique with honey bee behavior optimization for load balancing in cloud system environment,” Wireless Communications and Mobile Computing, vol. 2022, pp. 1–14, 2022.

[2] Z. Chen, “Research on Internet Security Situation Awareness Prediction Technology Based on Improved RBF Neural Network Algorithm,” Journal of Computer and Cognitive Engineering, vol. 1, no. 3, pp. 103–108, Mar. 2022, doi: 10.47852/bonviewJCCE149145205514.

[3] K. Shinan, K. Alsubhi, A. Alzahrani, and M. U. Ashraf, “Machine Learning-Based Botnet Detection in Software-Defined Network: A Systematic Review,” Symmetry, vol. 13, no. 5, p. 866, May 2021, doi: 10.3390/sym13050866.

[4] S. Namasudra, R. G. Crespo, and S. Kumar, “Introduction to the special section on advances of machine learning in cybersecurity (VSI-mlsec),” Computers & Electrical Engineering, vol. 100, p. 108048, May 2022, doi: 10.1016/j.compeleceng.2022.108048.

[5] A. Gutub, “Boosting image watermarking authenticity spreading secrecy from counting-based secret-sharing,” CAAI Transactions on Intelligent Technology, vol. 8, no. 2, pp. 440–452, Jun. 2023, doi: 10.1049/cit2.12093.

[6] S. Das and S. Namasudra, “Multiauthority CP-ABE-based Access Control Model for IoT-enabled Healthcare Infrastructure,” IEEE Transactions on Industrial Informatics, vol. 19, no. 1, pp. 821–829, Jan. 2023, doi: 10.1109/TII.2022.3167842.

[7] B. Bhola et al., “Quality-enabled decentralized dynamic IoT platform with scalable resources integration,” IET Communications, 2022.

[8] A. A. Laghari, A. A. Khan, R. Alkanhel, H. Elmannai, and S. Bourouis, “Lightweight-BIoV: Blockchain Distributed Ledger Technology (BDLT) for Internet of Vehicles (IoVs),” Electronics, vol. 12, no. 3, p. 677, Jan. 2023, doi: 10.3390/electronics12030677.

[9] M. Waqas et al., “Botnet attack detection in Internet of Things devices over cloud environment via machine learning,” Concurrency and Computation: Practice and Experience, vol. 34, no. 4, p. e6662, Feb. 2022, doi: 10.1002/cpe.6662.

[10] A. A. Laghari, X. Zhang, Z. A. Shaikh, A. Khan, V. V. Estrela, and S. Izadi, “A review on quality of experience (QoE) in cloud computing,” Journal of Reliable Intelligent Environments, vol. 10, no. 2, pp. 107–121, Jun. 2024, doi: 10.1007/s40860-023-00210-y.

[11] P. Rani and R. Sharma, “Intelligent transportation system for internet of vehicles based vehicular networks for smart cities,” Computers & Electrical Engineering, vol. 105, p. 108543, 2023.

[12] A. Wani, R. S., and R. Khaliq, “SDN-based intrusion detection system for IoT using deep learning classifier (IDSIoT-SDL),” CAAI Transactions on Intelligent Technology, vol. 6, no. 3, pp. 281–290, Sep. 2021, doi: 10.1049/cit2.12003.

[13] P. Rani et al., “Federated Learning-Based Misbehavior Detection for the 5G-Enabled Internet of Vehicles,” IEEE Transactions on Consumer Electronics, vol. 70, no. 2, pp. 4656–4664, May 2024, doi: 10.1109/TCE.2023.3328020.

[14] F. Sattari, A. H. Farooqi, Z. Qadir, B. Raza, H. Nazari, and M. Almutiry, “A Hybrid Deep Learning Approach for Bottleneck Detection in IoT,” IEEE Access, vol. 10, pp. 77039–77053, 2022, doi: 10.1109/ACCESS.2022.3188635.

[15] N. Hussain and P. Rani, “Comparative studied based on attack resilient and efficient protocol with intrusion detection system based on deep neural network for vehicular system security,” in Distributed Artificial Intelligence, CRC Press, 2020, pp. 217–236.

[16] P. Rani and R. Sharma, “IMFOCA-IOV: Intelligent Moth Flame Optimization based Clustering Algorithm for Internet of Vehicle,” in 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), IEEE, 2023, pp. 1–6.

[17] S. Mirjalili and A. Lewis, “The whale optimization algorithm,” Advances in Engineering Software, vol. 95, pp. 51–67, 2016.

[18] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey Wolf Optimizer,” Advances in Engineering Software, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[19] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, “Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems,” Advances in Engineering Software, vol. 114, pp. 163–191, Dec. 2017, doi: 10.1016/j.advengsoft.2017.07.002.

[20] H. Faris et al., “An efficient binary Salp Swarm Algorithm with crossover scheme for feature selection problems,” Knowledge-Based Systems, vol. 154, pp. 43–67, Aug. 2018, doi: 10.1016/j.knosys.2018.05.009.

[21] M. Tubishat, N. Idris, L. Shuib, M. A. M. Abushariah, and S. Mirjalili, “Improved Salp Swarm Algorithm based on opposition based learning and novel local search algorithm for feature selection,” Expert Systems with Applications, vol. 145, p. 113122, May 2020, doi: 10.1016/j.eswa.2019.113122.

[22] A. Elsaeidy, K. S. Munasinghe, D. Sharma, and A. Jamalipour, “Intrusion detection in smart cities using Restricted Boltzmann Machines,” Journal of Network and Computer Applications, vol. 135, pp. 76–83, Jun. 2019, doi: 10.1016/j.jnca.2019.02.026.

[23] Z. Baig, N. Syed, and N. Mohammad, “Securing the Smart City Airspace: Drone Cyber Attack Detection through Machine Learning,” Future Internet, vol. 14, no. 7, p. 205, Jun. 2022, doi: 10.3390/fi14070205.

[24] N. Hussain, P. Rani, N. Kumar, and M. G. Chaudhary, “A deep comprehensive research architecture, characteristics, challenges, issues, and benefits of routing protocol for vehicular ad-hoc networks,” International Journal of Distributed Systems and Technologies (IJDST), vol. 13, no. 8, pp. 1–23, 2022.

[25] M. Aloqaily, S. Otoum, I. A. Ridhawi, and Y. Jararweh, “An intrusion detection system for connected vehicles in smart cities,” Ad Hoc Networks, vol. 90, p. 101842, Jul. 2019, doi: 10.1016/j.adhoc.2019.02.001.

[26] M. Shafiq, Z. Tian, Y. Sun, X. Du, and M. Guizani, “Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for internet of things in smart city,” Future Generation Computer Systems, vol. 107, pp. 433–442, 2020.

[27] A. A. Elsaeidy, A. Jamalipour, and K. S. Munasinghe, “A Hybrid Deep Learning Approach for Replay and DDoS Attack Detection in a Smart City,” IEEE Access, vol. 9, pp. 154864–154875, 2021, doi: 10.1109/ACCESS.2021.3128701.

[28] T. Saba, A. R. Khan, T. Sadad, and S. Hong, “Securing the IoT System of Smart City against Cyber Threats Using Deep Learning,” Discrete Dynamics in Nature and Society, vol. 2022, no. 1, p. 1241122, Jan. 2022, doi: 10.1155/2022/1241122.

[29] N. Kumar Agrawal et al., “TFL-IHOA: Three-Layer Federated Learning-Based Intelligent Hybrid Optimization Algorithm for Internet of Vehicle,” IEEE Transactions on Consumer Electronics, vol. 70, no. 3, pp. 5818–5828, Aug. 2024, doi: 10.1109/TCE.2023.3344129.

[30] R. Vinayakumar, M. Alazab, S. Srinivasan, Q.-V. Pham, S. K. Padannayil, and K. Simran, “A Visualized Botnet Detection System Based Deep Learning for the Internet of Things Networks of Smart Cities,” IEEE Transactions on Industry Applications, vol. 56, no. 4, pp. 4436–4456, Jul. 2020, doi: 10.1109/TIA.2020.2971952.