ISSN: 2182-2069 / E-ISSN: 2182-2077

Cyber Security Identification Model Over an Ensemble Deep Learning System in IoT Networks

Boyella Mala Konda Reddy[1], Dr.A. Abdul Azeez Khan[2*], Dr.K. Javubar Sathick[3] and Dr.L. Arun Raj[4]

[1]Department of Computer Science and Engineering, B.S. Abdur Rahman Crescent Institute of Science and Technology, Chennai, Tamil Nadu, India. boyella_cse@crescent.education, https://orcid.org/0009-0002-0894-9352

[2*]Department of Computer Applications, B.S. Abdur Rahman Crescent Institute of Science and Technology, Chennai, Tamil Nadu, India. abdulazeezkhan@crescent.education, https://orcid.org/0000-0001-6960-752X

[3]Department of Computer Applications, B.S. Abdur Rahman Crescent Institute of Science and Technology, Chennai, Tamil Nadu, India. javubar@crescent.education, https://orcid.org/0000-0002-2248-8380

[4]Department of Computer Science and Engineering, B.S. Abdur Rahman Crescent Institute of Science and Technology, Chennai, Tamil Nadu, India. arunraj@crescent.education, https://orcid.org/0000-0001-8181-5022

Received: May 22, 2025; Revised: July 10, 2025; Accepted: August 11, 2025; Published: August 30, 2025

Abstract

Devices associated with the Internet of Things (IoT) are rapidly becoming ubiquitous, while IoT services are expanding their reach to become increasingly pervasive. IoT devices and applications are increasingly targets of attacks as a result of their rising popularity. IoT has long been a target of cyberattacks; however, as it becomes more pervasive in everyday lives and communities, people must take this threat with greater attention. Researchers have been trying to have a complete understanding of the threats and assaults that can harm IoT networks since safeguarding the IoT is a major issue. This work discovers the use of deep learning (DL) models in several facets of cybersecurity, analysing the results and finding parallels to machine learning (ML). An ensemble of Long Short-Term Memory (LSTM), Deep Convolutional Neural Network (DCNN), and Elman Neural Network (ENN) classifiers is used by the Multi-Class Classifier with Adaptive Horse Optimization (MCC-AHO) to identify the assault types. In this work, an ensemble technique known as majority voting was utilized, which involved optimizing the weights of the many classifiers to achieve the best possible result. This proposed model for cyber-attack detection assesses their effectiveness using the most recent CIC IoT Dataset 2023, achieving an accuracy of 98.16 %.

Keywords: IOT, Cyber Security, Deep Learning, CNN, Optimization, Long Short-Term Memory, Elman Neural Network.

Journal of Internet Services and Information Security (JISIS), volume: 15, number: 3 (August), pp. 539-559.
DOI: *10.58346/JISIS.2025.I3.037*

*Corresponding author: Department of Computer Applications, B.S. Abdur Rahman Crescent Institute of Science and Technology, Chennai, Tamil Nadu, India.

1 Introduction

Over the past decade, the Internet has expanded, making it simpler for people to find the facts, data, and services they need online (Lata & Kumar, 2022), while widespread access to the Internet has spurred the development and widespread use of the IoT (Grochowski et al., 2004). These days, the IoT is utilized in a wide variety of contexts, from smart homes and buildings to industries, medical, to smart businesses, to monitoring systems and even defence and infrastructure (Khattak et al., 2019; Bisong, 2019; Lin & Hsieh, 2015). This extensive adoption highlights the importance of continuing to evolve the IoT ecosystem to meet the future's mounting problems (Sikarwar et al., 2025). Performance and processing limitations, the varied features of IoT campaigns, and, most importantly, privacy concerns, security, and protocol-related risk factors are all obstacles that must be overcome.

The use of an Intrusion Detection System (IDS) is a reliable method of spotting malicious activity on a network. Most modern intrusion detection systems rely on ML algorithms for attack learning and detection. By strategically positioning distributed nodes closer to the items that make up the IoT network, fog computing is able to address the shortcomings of centralized Cloud computing in the areas of adaptability bottlenecks, greater bandwidth consumption, QoS (Quality of Service), and increased latency. Since fog-to-node computing moves computation, legitimacy, and processing of IDS closer to IoT network devices, it is ideal for the real-world installation and long-term viability of IoT networks (Purnama et al., 2024). Cyber-attacks are detected more rapidly and efficiently by IDS at fog nodes than in the cloud. Because of the wide variety of connected smart objects in IoT networks, from supercomputers to tiny devices with possibly little computation capacity, cybersecurity is a major weakness in the deployment of IoT networks (Aliero et al., 2020). In a distributed denial of service (DDoS) attack, hackers use an extensive number of hosts to overload a target server, causing the server to crash and making it impossible for legitimate users to access the service it provides (Chadd, 2018). This type of cyber-attack has caused significant damage to the IoT network in recent years. By 2023, there will be 17 million DDoS attacks per year, according to data from (Whitmore et al., 2015).

The IoT was conceptualized to enhance human well-being by connecting disparate smart devices and software platforms (Boopathy et al., 2025). However, there is still a significant obstacle in the way of assuring security in IoT contexts, and that is the wide variety of security risks that devices encounter. Many methods have been presented to ensure the safety of IoT gadgets, but there is always potential for development. One possible approach is to make use of ML, which has proven its capacity to spot patterns even when more conventional approaches have failed (Ayesh, 2024). IoT security can be improved with the help of DL, a branch of ML that uses larger deep neural networks for unsupervised, supervised, and semi-supervised learning. In (Rizk et al., 2019), a deep belief network was used to present the idea of DL, which has since been recognized to be operative in many other areas, including feature analysis, Natural Language Processing (NLP), and autonomous vehicles, among others. However, DL approaches require massive amounts of training data to achieve desirable results, which inevitably increases the training time. To address this issue, furthermore, describe a Lambda-based ensemble DL method that utilizes swarm intelligence (SI) for optimization in IoT settings (Heidari et al., 2019; Novak & Vacek, 2023). Cyber-attack categorization and enhanced cipher security are two of its primary applications. The swarm of particles is commonly utilized in SI algorithms; if one particle is missed during detection and comparison, the others will fill in the gaps. Using inter-particle communications, the particles determine the correct answer. Particles in motion and communication with one another throughout (Azizyan et al., 2019) various tasks might be thought of as a "swarm." SI learns how to solve problems by seeing how

species interact with one another socially. Numerous studies (Li et al., 2015; Li et al., 2017) have focused on optimization techniques (Elsayed et al., 2023).

Furthermore, built a hybrid IDS that can tell the difference between benign network traffic and five distinct types of attack traffic in real-time. The system takes a two-staged approach to processing the real-time traffic: first, a faster binary classifier distinguishes between regular and attack traffic; and secondly, the attack traffic gets divided into distinct types of attacks. Both of these classifiers are trained with DL techniques. In contrast to the superior performance of binary classifiers using LSTM, deep and hybrid ensemble-based techniques were utilized to enhance the accuracy of multi-classification. Both accuracy and throughput are measured and analysed to determine performance levels. The results show that the multi-staged approach improves accuracy and throughput by separating the decision-making for real-time tasks from the analysis and training of the models.

The rest of this research paper is structured as follows: The literature on IDS in the IoT is presented in the "Related work" section. The "Proposed framework" section summarizes the suggested MCC-AHO design framework and batch data classification for detecting attacks in the IoT context. After conducting experiments using the proposed method and utilizing the most common assessment criteria, we present these findings in the "Performance Analysis" section. The last "Conclusion and future work" section presents final thoughts.

2 Related Work

This discussion is around literature that has concentrated on the detection of network intrusions in the IoT through the utilization of ML and DL methodologies. In their study, Martins et al., (2022) introduced an ensemble methodology that leverages ML techniques to effectively identify attacks in the IPv6 Routing method for low-powered and fragile IoT (Kumar, 2024). The strategy demonstrated a notable level of accuracy in its detection capabilities. The ensemble-based technique provided by Pan et al., (2022) utilizes Deep Belief Networks to address the issue of redundant features. They tackle this problem by employing a Minimized Redundancy Discriminative Feature Selection method. Ahmad and colleagues (Ahmad et al., 2022) proposed a novel ensemble DL methodology that integrates DNN (Deep Neural Network) and DT (Decision Tree) classifiers to discern attack types without relying on a binary classification framework. Sarker et al., (2023) employed an ensemble methodology that relied on conventional ML techniques rather than DL methodologies to identify network breaches in the context of the IoT. In their study, Malik et al., (2022) proposed an ensemble-based method for anomaly detection, specifically targeting DoS (Denial of Service), DDoS (Distributed Denial of Service), reconnaissance, and key-logging assaults. Their methodology involved the integration of a One-Class Support Vector Machine and C5 classifier.

In their study, Mehedi et al., (2022) employed LSTM and CNN models, in conjunction with character encoding, to explore SFL (Spatial Feature Learning) strategies for distinguishing between obfuscated and encrypted HTTP traffic and regular traffic. It is worth noting, however, that their approach does not involve ensemble methods and primarily concentrates on obfuscated packets alone. In their study, Ghimire & Rawat, (2022) introduced an enhanced methodology based on CNN and LSTM networks for the identification of Attacks DDoS (Distributed Denial of Service) on social media platforms. Their approach showed superior performance compared to the conventional execution of LSTM and CNN. The current study does not encompass other notable instances of cyber-attacks and does not employ the Lambda construction for typicaltesting andtraining. In their study, Khan et al., (2022) conducted a comprehensive examination of different DL algorithms in the context of IoT environments. This analysis

was carried out utilizing diverse network-based datasets. In their study, Otoum et al., (2022) introduced a novel IDS for IoT devices. The system utilizes a DL approach and incorporates the SMO (Spider Monkey Optimization) algorithm for optimal feature selection. Additionally, the system employs the SDPN (Stacked-Deep Polynomial Network) for classification purposes. The researchers conducted detection techniques for many types of cyber-attacks, including DoS (Denial of Service), remote-to-local (R2L), probe, and user-to-root (U2R).

In their study, Alenezi et al., (2021) introduced a methodology that utilizes ML and DL techniques to recognize DoS attacks within IoT networks. The technique projected by Docs (Ackoski, 2020) involves the utilization of a SMO algorithm in conjunction with SDPN to attain optimal intrusion detection. The authors Idrissi et al., (2021) proposed a DL approach that utilizes CNN to identify and detect Botnet attacks. The authors asserted that their approaches yield results with an accuracy of 99.04% and a forecast time of 0.34 milliseconds. The Lambda architecture is a structured methodology for efficiently handling large volumes of data through a combination of batch and stream-processing techniques. Researchers have utilized the Lambda architecture to provide IDS solutions for IoT contexts that generate large amounts of data. The aim is to reduce implementation and computing costs while improving efficiency. According to Lahasan & Samma, (2022), the full potential of the Common Vulnerability Scoring System (CVSS) classification approach, as indicated by Lal et al., (2023), has not been fully realized in terms of its implementation and its relevance for evaluating security in application packages. The research demonstrates that the suggested approach for ensuring mental security is more effective in resolving the aforementioned deficiencies and the challenges posed by the diversity of cybersecurity alert systems. Ahmad et al., (2021) have presented a comprehensive analysis that examines the utilization of diverse ML and DL techniques for the identification of distinct categories of attacks. Moreover, this study delves into the examination of the diverse platforms and tools employed in the implementation of DL and ML techniques. Additionally, it provides a concise overview of the security measures devised to counteract the various types of assaults within each category.

The majority of study endeavors concentrate on singular or restricted IoT attacks, hence neglecting a comprehensive range of IoT attack scenarios. The majority of existing studies primarily depend on the utilization of individual binary classifiers for distinguishing between malicious and benign packets, hence lacking the capability to effectively identify the specific types of assaults targeting IoT devices. These works do not utilize ensemble-based methodologies to enhance accuracy and precision. The suggested research work solves all these concerns by proposing a deep ensemble IDS technique with swarm intelligence-based optimization schemes.

3 Proposed Methodology

In this context, the author emphasizes the need to employ information storage, data processing, and data input systems in constructing the Lambda framework. This architecture is designed to effectively manage both streaming and batch data for the goal of classifying IoT device assaults. Furthermore, the author suggests utilizing DL methodologies to enhance the accuracy of attack detection within this framework. Additionally, this paper addresses the topics of data preparation, feature extraction, training procedures, and the process of attack detection and classification. The representation of the projected MCC-AHO is demonstrated in Figure 1. The first step in the data analysis process involves pre-processing the data to eliminate any irrelevant or undesired information. Subsequently, the salient characteristics are extracted from the dataset. The data is organized utilizing the Matthews Correlation Coefficient (MCC) in conjunction with the AHO method to distinguish between regular and assault

traffic (Chicco & Jurman, 2020). The effectiveness of the suggested scheme was tested in comparison to existing classification schemes. The results indicate that the suggested approach achieved favourable outcomes, attributed to its effective classification and optimization techniques.

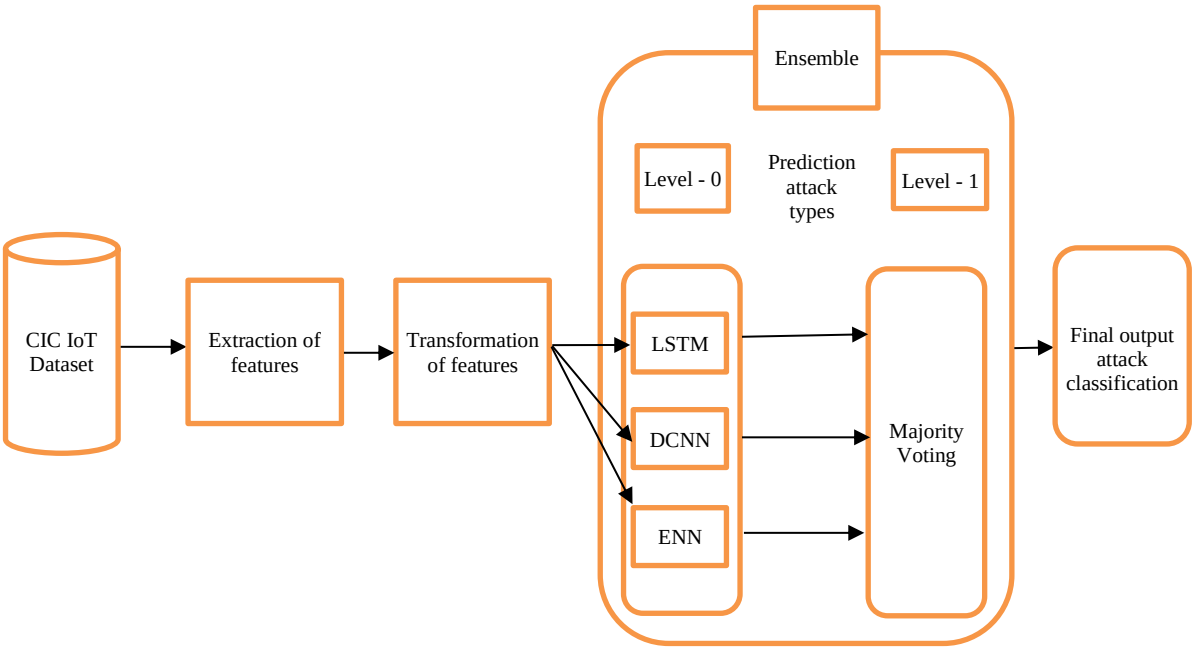

Figure 1: The Complete Architecture of the Proposed Mcc-Aho Classification

Data Preprocessing

To address the issue of imbalanced data as well as ensure equitable outcomes across every group, the researchers employed a technique known as random under-sampling. This approach mitigates the possibility of bias in the evaluation or ML algorithm by reducing the influence of the majority class. During the training process of machine and DL algorithms, furthermore utilized Minmax Scaling inside the range of [0,1] to enhance the data quality and expedite the convergence of the trained models.

Lambda Architecture

The Lambda architecture offers a systematic approach to managing and analyzing vast quantities of data, rendering it an appropriate architectural resolution for addressing IoT situations. The routine of this architectural framework enhances the system's capacity for high availability, flexibility, scalability, and adaptability. Figure 2 explains the fundamental functioning of the lambda framework. The data-processing architecture in question employs a combination of batch and stream processing techniques to effectively handle substantial volumes of data. This method of design tries to achieve an equilibrium between latency, fault tolerance, and throughput by utilizing batch processing to offer comprehensive and precise perspectives on group data, and instantaneous stream analysis to deliver insights into live data. It is feasible to amalgamate the two perspective outputs before their presentation, handling massive volumes of data, and minimizing map latency.

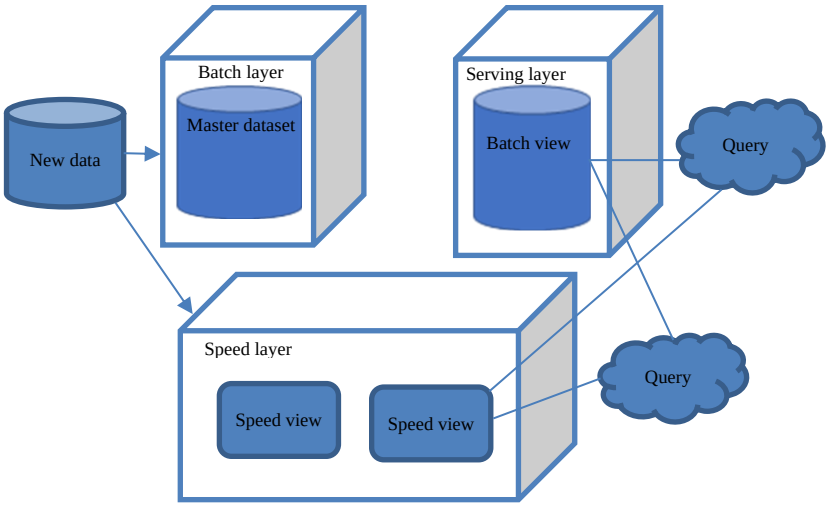

Figure 2: Architecture of Lambda for Preprocessing

Feature Selection

The process of feature selection from network data is a crucial stage that has a direct impact on the effectiveness of the classifiers. The aforementioned aspects exhibit a profound reflection of the inherent qualities of network traffic, encompassing intricate details and distinct attributes of both benign and malicious network-related activities. The process of choosing features for a binary classifier differs from picking network characteristics for a multi-classifier since in the latter scenario, distinct features contribute to the identification of various types of network intrusions. For example, a particular attribute may exhibit significant use in discerning Denial-of-Service (DoS) attacks, yet demonstrate limited efficacy in identifying malevolent cyberattacks. Hence, the network features chosen must be directly associated with the specific sort of attack being investigated. In the context of identifying anomalous traffic, it is important to note that not all network parameters are equally influential. Certain network features may have limited or negligible utility in discerning unusual traffic patterns. Given the presence of numerous categorical features in the dataset, it was necessary to perform feature transformation, as DL models excel in handling numerical data. To achieve this objective, the Fit Transform method was employed, which is readily accessible in the Scikit-Learn Python package. The encoder allocates values within the range of 0 to n-1, wherein n is the entire count of records being taken into account.

Batch layer: The batch layer in the lambda model is characterized by its high latency and great accuracy. It is specifically designed to handle data in batches rather than in real-time. Moreover, the aforementioned properties of the batch layer have been employed to train both classifiers within the batch layer, utilizing a publicly accessible benchmark dataset. In this context, it is worth noting that concerns regarding latency and time on the training of these classifiers are not present. The comprehensive training procedure at the batch layer is depicted in Figure 2. To train the binary and multi-class classifiers, three DL classifiers were employed, such as ENN, DCNN, and LSTM.

DCNN Model: The CNN is composed of one or more convolutional layers, which are subsequently connected by several layers that are fully connected. The output and input layers are interconnected via numerous hidden layers, often consisting of a series of convolutional layers. CNNs exhibit proficiency in the identification of malware attacks, harmful scripts, and codes, as well as the detection of anomalous traffic patterns. However, the computing time was found to be very high. To mitigate this issue, the hyperparameter was optimized utilizing the AHO technique.

Layer 1 of a CNN represents the initial convolutional layer within the model used to generate an array of features. A convolutional operation is conducted on the input feature matrix.

The first layer of Maxpooling. The dimensions of the CNN layer1 features are diminished throughout their propagation to this particular layer, resulting in a reduction in the filters' susceptibility to noise and fluctuations.

The second layer of a CNN operates similarly to the first layer, with the distinction being that the initial layer gathers low-level characteristics from the data, whereas the subsequent convolutional layer extracts high-level features.

In the second layer of Maxpooling, the primary objective remains the same as in the first Maxpooling layer, which is to diminish the dimensionality of the feature map. This process results in the generation of a feature pool array.

The flattening operation is executed on the matrix derived from the 2nd Maxpooling layer. The conversion of the pooled feature array into a feature vector is performed by this layer, resulting in a column or vector representation.

Classification is performed using the feature vector obtained from the flattened layer, in conjunction with the utilization of softmax activation algorithms. Ultimately, the traffic nodes undergo an examination to classify potential assaults. The fundamental rationale for employing CNN as the proposed technique stems from its effectiveness in feature identification and recognition tasks. In DCNN, the integration of several activation functions such as ReLU (Rectified Linear Unit) and Hard-Swish has been observed to result in neural networks that exhibit enhanced performance.

Convolution 1: The source traffic features consist of a matrix of numerical features and are prepared for further processing through a convolutional layer. Equation (1) is utilized for the computation of the feature matrix Mat, which is defined as follows:

$$Mat = Cd(ix) \cdot (N_w, \mathfrak{b}i \times I_{r+\mathfrak{w}-1,c+h-1} + \mathfrak{b}i) \quad (1)$$

In this context, the term "CD-Channel dimension" refers to the dimensions of a matrix. The variable "r" represents the row of the matrix, which ranges from 1 to the value of "l". Similarly, the variable "c" represents the column of the matrix, which ranges from 1 to the value of "h". The variable "N" represents the number of features, denoted as "k", ranging from 1 to "n". The term "fn(x)" represents an activation function applied to the numeric feature "I". Lastly, "b" represents the bias term. In this study, a nonlinear activation function was employed. Specifically, the ReLU activation function was chosen due to its strong expressive capacity and its capability to mitigate the vanishing gradient problem, hence ensuring the stability of the model's convergence rate. The function fn(x) represents the ReLU formula, as denoted by Equation (2). When the variable ix is an integer,

$$relu\big(fn(ix)\big) = \begin{cases} 0 & if\ x \le 0 \\ x & if\ x > 0 \end{cases} \quad (2)$$

This study presents the introduction and definition of a unique activation function called Hard-Swish (R), as represented by Equation (3).

$$\big(fn(ix)\big) = 2\,x \times \max\big(0, \min\big(1, (\alpha x \times 0.2 + 0.5)\big)\big) \quad (3)$$

Where α is a variable that can be either a parameter that can be trained or a constant. As the value of α tends towards infinity, the hard-sigmoid component tends to approach the range of 0 to 1. Consequently, the behavior of the function R becomes similar to that of the Hard-Swish (R) activation function. The parameter α can be utilized as a trainable variable to regulate the level of interpolation

inside the model. Therefore, it is suggested to combine various activation functions, namely in the channel dimension, by concatenating them.

$$Cd(ix) = ct\left(\left[R\left(fn(ix)\right), relu\left(fn(x)\right)\right]\right) \quad (4)$$

As expressed in Equation (4), the variable "ct" represents the concentration.

Maxpooling 1: To mitigate the issue of overfitting and reduce the number of parameters in lower networks, the utilization of a Max pooling layer becomes crucial. This layer effectively eliminates irrelevant features by performing downsampling operations.

$$ix_j^l = fn\left(w_j^l * d\left(ix_j^{l-1}\right) + \mathfrak{b}i_j^l\right) \quad (5)$$

The weight of the jth feature map in the lth layer is denoted as w_j^l, whereas the bias in that map is represented by $\mathfrak{b}i_j^l$. In the field of statistics, down-sampling refers to the utilization of various functions such as the mean, maximum, and stochastic pooling functions. The technique of employing shift-invariant max-pooling was employed to decrease the dimensionality of the output feature map.

The second convolutional layer is active to extract higher-level features from the input, which is the pooled functional matrix obtained from the Maxpooling layer. The addition of the second convolution layer is equivalent to that of the first convolution layer, as described by equations 1 and 2.

The objective of the second max-pooling layer is to decrease the size of the matrix, and it is calculated similarly to the initial max-pooling layer as in Equation (5).

The Flatten Layer is responsible for processing the output of the second max pooling layer, which is a pooled feature map. Its objective is to transform a column or feature vector from a pooled feature matrix. The features or parts of the pooled feature map $Im$are are transformed into feature vectors Iv within this layer by a process of function restructuring, as outlined in Equation (6).

$$Iv = p.r\left[\left(relu\left(fn(xi)\right) - w + 1\right) \times \left(R\left(fn(xi)\right) - h + 1\right)\right] \quad (6)$$

The variable p represents the pooling operation, while the variable r represents the reshape operation.

Classification involves the calculation of probability for different kinds of attacks by managing a dense layer with numerous neurons using softmax algorithms. Therefore, the net output Y can be derived using the following equation: Equation (7)

$$Y_j = \sum_{i=1}^{l} w_i \cdot xi_i + \mathfrak{b}i \quad (7)$$

The softmax activation functions are employed at the classification layer.

Hyperparameter tuning: The present AHO study presents an efficient approach for conducting parameter search in the proposed CNN. Hyperparameter tuning necessitates the specification of several parameters, including but not limited to kernel size, number of steps, number of channels, weight, and bias. Hyperparameter tuning facilitates the identification of an optimal parameter configuration for a model, hence maximizing its performance outcomes. This study uses Hyperparameter Tuning to determine the optimal parameter selection by comparing the recommended parameters using AHO and a modified structure known as DCNN.

LSTM Model: The LSTM is a type of recurrent neural network (RNN) that can detect and learn from causal relationships over time. The layers interact in a unique way and help find state maintenance-related attacks like DDoS, cross-site scripting, and other kinds of application-focused attacks. Utilisation of the gate structure has the potential to enhance the transmission of information and facilitate

communication among cells, while successfully mitigating the challenges associated with the vanishing and exploding gradients commonly observed in traditional networks. The performance of LSTM is greatly impacted by both the underlying assumptions of the model and the architecture of the network, in addition to the former. The achievement of optimal efficiency in the model is hindered by the inherent limitations of traditional feature identification methods, which are susceptible to human biases.

Consequently, this discrepancy between predicted accuracy and the desired level of accuracy arises. The AHO strategy has the potential to increase the performance of the model by successively identifying the optimal variables for the LSTM model. This approach significantly diminishes the reliance on human experience in determining the model's output results. Because of this, the present study employs a method to update the internal weights of LSTM using the Weight Optimization (WO) technique. The study iteratively identifies the optimal parameters for the model, constructs the LSTM model using the ideal configurations to extract temporal information from electrocardiogram (ECG) data, and subsequently predicts arrhythmia framework points based on the ECG signals. The framework of the LSTM-rBM, which is a modified version of the RNN-rBM (rBM with Recurrent Neural Network), exhibits similarities to those of the RNN-rBM (Azumah et al., 2021). In addition, the RBM can be employed to elucidate the temporal features of the sequences. This involves establishing dynamic connections between the hidden layer of the LSTM and both the visible and hidden elements of the RBM. The two biases of the rBM, namely $bs_{hi}^{(t)}$ and $bs_{vi}^{(t)}$, are updated using the temporal data obtained from the LSTM model. The update mechanism for these biases follows the same methodology as described in reference (Nakashika et al., 2014).

The RNN and LSTM models include distinct internal mechanisms. In the LSTM, the current input $x^{(t)}$is considered with the previous hidden layer output $op^{(t)}$ and the current output $vi^{(t)}$. The visible layer of the RBM has a crucial role in influencing the current output of the LSTM model. Additionally, the descriptions of the renewal equations for each output gate (op_t), output state (C_t), forget gate (fg_t), and internal input gate (ip_t) of the Cells in the LSTM are determined by utilizing Equation (8) mentioned in reference (Zhuo et al., 2017).

$$hi_t = op_t * \tanh(C_t) \quad (8)$$

To enhance the sample efficiency inside the RBM component of the model, both Gibbs sampling and Persistent Contrastive Divergence (PCD) techniques are employed.

ENN Model: The ENN is a type of feedback neural network (FBNN) that incorporates a context layer, serving as a delay operator, within the hidden layer of a backpropagation (BP) neural network. This addition of a context layer allows the network system to effectively adapt to time-varying variable features and exhibit robust global stability (Hafez et al., 2015; Wang & Wang, 2019; Aldeen et al., 2014). The field of topology is commonly categorized into four distinct levels, including the hidden layer (second layer), context layer (third layer), input layer (first layer), and output layer (final layer). The context layer is employed to store the output of the hidden layer, serving as a mechanism akin to a step delay operator. The output of the hidden layer in a BP network is connected to its input through the delay and storing mechanism of the context layer. This approach to relationship management exhibits sensitivity towards past data, while the incorporation of an internal feedback network enhances the capacity to effectively handle current information. The inclusion of the internal state enables the system to possess a dynamic mapping function, hence allowing it to respond to time-varying features.

Assuming an input of "in" and an output of "not," the neural network consists of no hidden neurons and context neurons. The weights relating the input layer directly to the corresponding hidden

layer are denoted as wt_1, the weights concerning the context layer to the hidden layer are denoted as wt_2, and the weights connecting the hidden layer to the output layer are denoted as wt_3. The input of the neural network at time step (k-1) is signified as $ui(k-1)$, the output of the hidden layer at time step k is represented as $xh(k)$, the output of the context layer at time step k is represented as $xt_c(k)$ in Equation (9), and the productivity of the neural network at time step k is represented as $yn(k)$.

$$xh(k) = fn\left(wt_2 xt_c(k) + wt_1\left(ui(k-1)\right)\right); xt_c(k) = xi(k-1) \quad (9)$$

The hidden layer transfer function in Equation (10), typically denoted as fn, is frequently employed in S-type functions.

$$fn(x) = (1 + e^{-x})^{-1} \quad (10)$$

The transfer function of the output layer in Equation (11), denoted as g(t), is typically linear.

$$yn(k) = gt\left(w_{t3} xh(k)\right) \quad (11)$$

The Error Neural Network (ENN) employs the Backpropagation (BP) method to adjust the weights as represented in Equation (12).

$$ENN = \sum_{k=1}^{m} (t_k - yn_k)^2 \quad (12)$$

The output vector of the object is represented by the variable Wheret_k. To enhance the accuracy of the ENN prediction, the weight value of the ENN is improved by the utilization of the AFO algorithm.

Hyperparameter Tuning All Deep Classifiers Using AHO

The AHO Algorithm draws primary inspiration from the structured hierarchy of herds of horses. Like many other animals that have adapted to communal living, horses benefit from a well-established "pecking order" within the herd to decrease conflict and foster solidarity among members. In a non-linear structure, horse P may be prominent over horse Q, who may be dominant over horse S, and horse S may be dominant over horse P. This is usually, but not always, a linear system. Several aspects contribute to dominance, such as the urgency with which one requires a resource. As a result, it can change throughout the lifespan of the herd or of the individual animal. Horses can either be completely dominating or completely submissive concerning access to resources. This is not normal horse behavior, but rather the result of humans cramming too many horses into too little space with too few luxuries. Many so-called "dominant horses" really have impaired social skills. There is a clear pecking order among horses in a herd, with the dominant animals receiving food and water first. If there is not enough food to go around, the horses with better social standing may prevent the lower-status horses from eating altogether.

Each herd of horses has a head stallion or mare who controls the distribution of food and other resources. In the first stage of the algorithm, the health scores of the horses in a herd are used to determine the order of the horses in that herd, as in Equation (13). As given in Equation (14), let F be a function, and herd k is a set of horses.

$$Herd = \{Hr_1, \ldots, Hr_k\} \quad (13)$$

$$F = Herd \rightarrow \{1 \ldots k\} \quad (14)$$

Where fit is fitness and $x \neq y, x \,\&\, y \in \{1 \ldots k\}$ are integers, if $fit\,(Hr_x) < fit(Hr_y)$, then Equation (15) presents as follows.

$$\left[F(Hr_x) - F\left(Hr_y\right)\right](x - y) > 0 \quad (15)$$

The Hr_x ranking of each horse is determined by using Equation (16).

$$Hr_x - each\ horse\ rank = \frac{F(Hr_x)}{k} \tag{16}$$

Because rankings (r) of horses signify relative importance, the center (c) of a herd is equal to the weighted average of the location of the horse within the herd. Locating the middle of a group of animals involves using Equation (17).

$$Herd_c = \frac{\sum_{x=1}^{k} \mathrm{Z}_x Hr_{x,r}}{\sum_{x=1}^{k} Hr_{x,r}} \tag{17}$$

The Euclidean distance is used to calculate the distance between the location of the stallion and the center of the horse herd, which is denoted by Hr as in Equation (18).

$$D(S, Herd) = \sqrt{\sum_{y=1}^{D} \left(S_y - Hr_c\right)^2} \tag{18}$$

Here D is dimensions in the search space, S = stallion. If the horse joins the specified group, it will adjust its speed by Equations (19) and (20).

$$Vl_{x,y}^{Ti+1} = Vl_{x,y}^{Ti} + Hr_{x,r} * \left(Herd_{c,y}^{Ti} - \mathrm{Z}_{x,y}^{\mathrm{Ti}}\right) \tag{19}$$

$$Vl_{x,y}^{Ti+1} = Vl_{x,y}^{Ti} + R * \left(Herd_{c,y}^{Ti} - \mathrm{Z}_{x,y}^{\mathrm{Ti}}\right) \tag{20}$$

Ti is the current iteration, and $Ti + 1$ is the next iteration, where R is a random number between 0 and 1. As denoted in Equation (21), Mr is a matrix with as many rows as MHP and as many columns as Dc, where MHP is the size of the horse memory pool.

$$Mr_x^{Ti+1} = \begin{bmatrix} Mr_{1,x,1}^{Ti+1} & \cdots & Mr_{1,x,Dc}^{Ti+1} \\ \vdots & \ddots & \vdots \\ Mr_{MHP,x,1}^{Ti+1} & \ldots & Mr_{MHP,x,Dc}^{Ti+1} \end{bmatrix} \tag{21}$$

To refresh the memory matrix cells, Equation (22) is used.

$$Mr_{k,x,y}^{Ti+1} = \mathrm{Z}_{x,y}^{\mathrm{Ti}} * N\ (0, stdD) \tag{22}$$

Normal distribution (N) is defined as follows: N = mean 0 + stdD. To speed up convergence and enhance HO. An adaptive scaling factor (ASF) is used in the AHO algorithm to improve the efficiency of the HO procedure. Using the following equation, furthermore, can determine the ASF as represented by Equation (23):

$$ASF = 2 - \min\left\{2, \frac{OP_*^{Ti-1}}{OR_*^{Ti}}(1 - \cos\frac{Ti\pi}{t}\right\} \tag{23}$$

Where O is the set of fitness maxima, R is the most recent iteration, and O(t-1) is the set of fitness maxima from the iteration before it. Non-linearity is represented by the cos function. It has enhanced performance precision and reached zero or one state. Algorithm 1 depicts the entire process of tuning parameters and the flowchart given in Figure 3.

Algorithm 1: HOA For Parameter Tuning in Deep Classifiers

Input: weight parameter values

Output: optimal weight selection (i.e., optimal fitness)

1. Set the population of horses (i.e. Weight value) in the search space

2. Find fitness and update to the best one
3. While iteration <Ti
4. Determine the center position of the horse from Ti
5. Ranking the house from top to bottom
6. If fitness condition is satisfied
7. End
8. Else, go to step 2
9. Ti=Ti+1
10. Compute the global best solution
11. Return the global best solution as the hyperparameter value

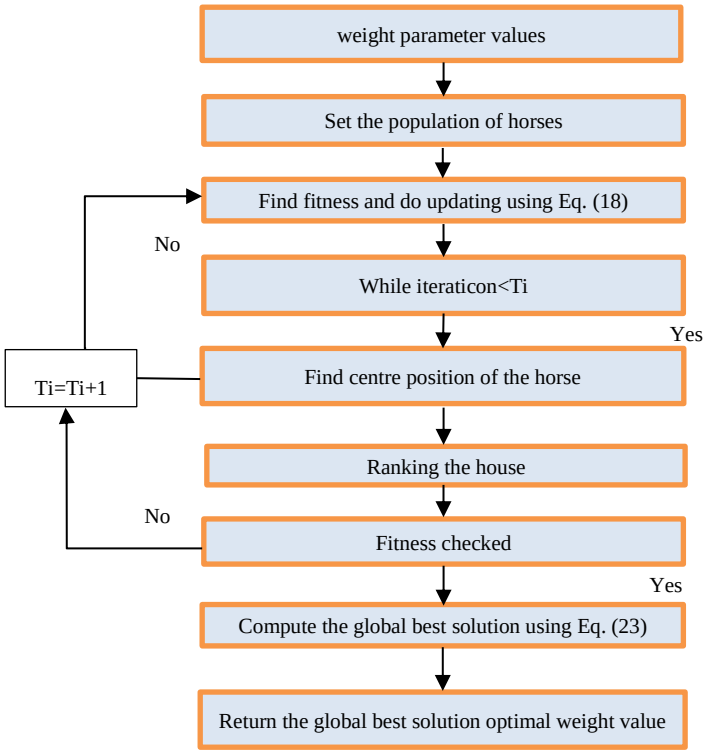

Figure 3: Flowchart of HOA for Parameter Optimization of MCC

Binary classifier: Additionally, all three DL classifiers were applied to the training dataset separately before the binary classifier in the batch layer was trained. The LSTM binary classifier was also found to be superior to DCNN and ENN-based binary classifiers during testing, both in terms of accuracy and detection of attacks. This classifier is limited to determining if a packet on a network is safe or harmful; it cannot identify the specific form of assault. The ensemble-based multi-class classifier is responsible for determining the classification of any malicious packets. This method will help save valuable processing time by bypassing the ensemble-based multi-class classifier for benign packets, as the vast majority of network traffic in IoT contexts is benign in real time. Algorithm 2 demonstrates the inner workings of the Binary Classifier in detail.

Algorithm 2: Pseudocode of Binary Classifier

Input: Traffic stream $Tx = (Tx_1, Tx_2, .. Tx_2)$, η −extracted features,ϵ-transformed features, features (Tx), N←seek LSTM (ϵ)

Output: trafficdetection identification type N

Start binary class in Tx for classification

If (N=normal) then

Implement and endure

Else

Multiclass applied in Txfor classification

$$perdiction\ 1 \rightarrow \alpha\ for\ LSTM\ classification$$
$$perdiction\ 2 \rightarrow\ \beta\ for\ DCNN\ classification,$$
$$perdiction\ 3 \rightarrow \gamma\ for\ ANN\ classification$$

MA: prediction of majority voting (α, β, γ)

Represents the final output

End if

end

Deep ensemble multi-class classifier: Additionally, three separate classifiers—LSTM, CNN, and ENN—were first employed to train a classifier to do multi-class classification, and at this point, their hyperparameter tuning was also completed using AHO. However, LSTM showed the best results (98.20%) of any method tested. The algorithm demonstrates an ensemble method called Majority Voting, which was utilized to further enhance performance. The ultimate forecast is made by majority vote at Level 1 (as depicted in Figure 1), which is based on the abilities of the specific learners (known as basis learners at Level 0). Using this ensemble method, we were furthermore able to attain a 99.5% accuracy in the outcomes.

4 Performance Results and Discussion

In this context, furthermore, assess the performance of the proposed MCC-AHO and compare it to that of well-established ensemble algorithms such as LSTM, CNN, and ANN (Ghillani, 2022). The following metrics will be used to assess the outcome.

Dataset Description

The most recent CIC IoT Dataset (2023) (Neto et al., 2023) has been used in the proposed MCC-AHO effort. The primary purpose of this study is to present a fresh and comprehensive IoT attack dataset to encourage the growth of security analytics usage in operational IoT settings. This is done by carrying out 33 attacks on a 105-node IoT network. DoS, DDoS, Recon, Brute Force, Web-based, Spoofing, and Mirai are just a few of the names given to these various forms of cyberattack. Last but not least, every single assault is carried out by hostile IoT devices that specifically target other IoT devices. This dataset includes the newest available network data, both with and without an attack, and is designed to be as representative as possible of the network of a working organization. Since the imbalance of this dataset has a significant impact on the training of the DL method and, by extension, the testing, we have corrected the problem by creating duplicates of the data.

Evaluation Metrics

The suggested IDS, which uses DL models to identify attacks, was evaluated using the most important performance factors. The following are some of the most popular measures of quality used by scientists:

- Recall that this assessment metric evaluates the amount of actual positive instances that were forecast positive.
- Accuracy, often known as predictive value, is the rate at which one's predictions of either a favorable or negative outcome come to fruition. The concept of precision is widely used in fields such as data mining, ML, and information retrieval to characterize the performance model.
- Precision can be defined as the classification model's total performance.
- The F1-score, or F-measure, is a metric that takes both recall and precision into account.
- Throughput is the rate at which output is generated and evaluated successfully based on the quantity of flow.

The effectiveness of LSTM, DCNN, and ENN binary classifiers. Furthermore, separately deployed and tested the three DL models to compare and contrast how well they performed on the testing dataset while assessing the binary classifier. By calculating the accuracy and loss during model training and validation, we were able to arrive at the best possible binary classification model in terms of evaluation parameters. The results of the suggested MCC-AHO, ENN, DCNN, LSTM, and SCEEHO-SPC-CNN-CD-DBN-based classifiers for the binary classification model are summarized in Table 1. In terms of accuracy and precision, MCC-AHO is superior to other LSTM, ENN, DCNN and SCEEHO-SPC-CNN-CD-DBN-based classifiers. Indeed, in the end, we only used the MCC-AHO in the batch-trained binary classification model. The idea had significant success because of ensemble categorization.

Table 1: Binary Classification Evaluation Results

Model	Traffic type	accuracy	Precision	Recall	F1-score	Throughput (ms)
Proposed MCC-AHO	Normal	99.5	98.9	99.2	98.3	0.08
	Attack		99.56	98.7	97.8	
SCEEHO-SPC-CNN-CD-DBN	Normal	98.95	98.56	98.95	97.56	0.07
	Attack		99.42	98.32	96.5	
LSTM	Normal	98.2	93.6	97.3	96.3	0.065
	Attack		96.7	94.2	94.3	
DCNN	Normal	94.58	92.3	89.33	85.6	0.04
	Attack		93.4	95.7	90.43	
ENN	Normal	93.54	90.6	87.2	82.5	0.03
	Attack		91.3	90.3	86.7	

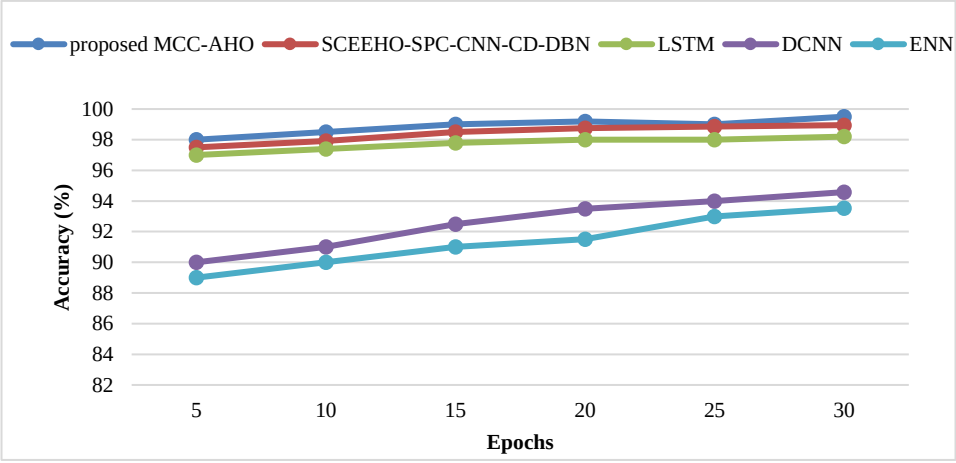

Figure 4: Accuracy Results Comparison of MCC-AHO, SCEEHO-SPC-CNN-CD-DBN, LSTM, DCNN, and ENN

Figure 4 displays the agreement between the proposed MCC-AHO and the already used models for a certain database's epoch count. The MCC-AHO shortens processing times without sacrificing precision. Because it does not necessitate a large number of epochs during reduction, the MCC-AHO achieves an accuracy of 99.5% compared to all other classification models like SCEEHO-SPC-CNN-CD-DBN, LSTM, DCNN, and ENN. Since the MCC-AHO network effectively avoids the risk of overfitting in short datasets and can more effectively learn and fit network traffic features, it outperforms the state-of-the-art algorithms in terms of good validation findings for attack prediction.

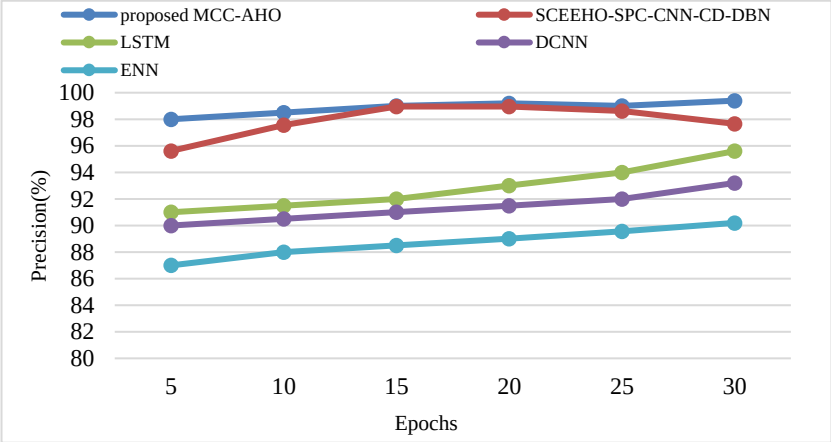

Figure 5: Precision Result Comparison of MCC-AHO, SCEEHO-SPC-CNN-CD-DBN, LSTM, DCNN and ENN

Figure 5 depicts the precision of the proposed MCC-AHO compared to other models, such as SCEEHO-SPC-CNN-CD-DBN, LSTM, DCNN, and ENN, for a precise set of database attributes. As the number of epochs grows, so does the accuracy. The MCC-AHO, for instance, improves upon previous methods by achieving a 99.4% precision rate. This is because MCC-AHO reduces the amount of time spent computing the derived factors, making it simpler to tune for optimal precision. In addition, utilizing a RELU activation mechanism for each text representation helps to zero in on the most important connected information while preventing the interference of one text representation with another. As a result, the accuracy of traffic classification and prediction is greatly enhanced by this approach.

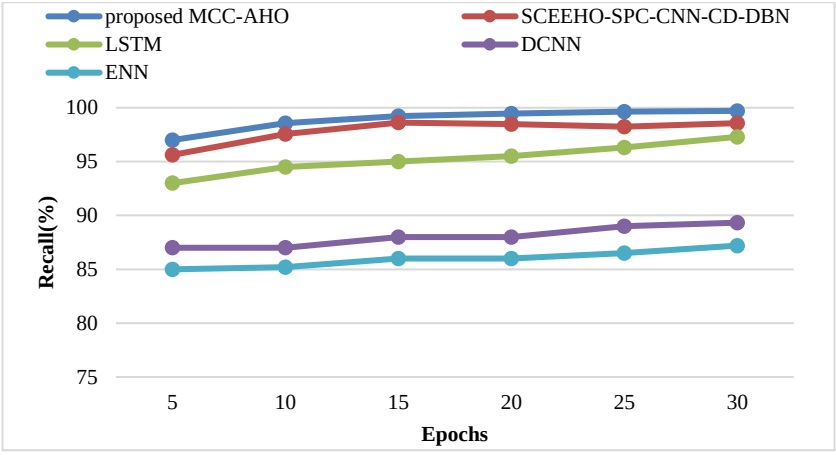

Figure 6: Recallresults Comparison of MCC-AHO, SCEEHO-SPC-CNN-CD-DBN, LSTM, DCNN, and ENN

Figure 6 depicts the Recall for various feature counts for the proposed MCC-AHO and other current models such as SCEEHO-SPC-CNN-CD-DBN, LSTM, DCNN, and ENN. The accuracy improves with the number of iterations, or epochs. The MCC-AHO, for instance, improves upon previous methods by reaching a recall of 99.2 %. This is because MCC-AHO reduces the amount of time spent computing the derived factors, allowing for more rapid fine-tuning and thus a higher recall rate. By minimizing processing time, AHO's optimal routing fine-tunes the effectiveness of ensemble categorization.

Compared to the current models SCEEHO-SPC-CNN-CD-DBN, LSTM, DCNN, and ENN, the suggested MCC-AHO has a higher F1-score for the number of features in the supplied databases (see Figure 6). The f-measure is optimized concurrently with the number of epochs. The MCC-AHO, in comparison to all other models, yields an f-measure of 98.3 %. As was seen above, the MCC-AHO fared better than the other approaches on the accuracy and F1 scale. When applied to WHO, these processes allow for significant improvements in hyperparameter quality for MNN at low computational cost, leading to a high F1score.

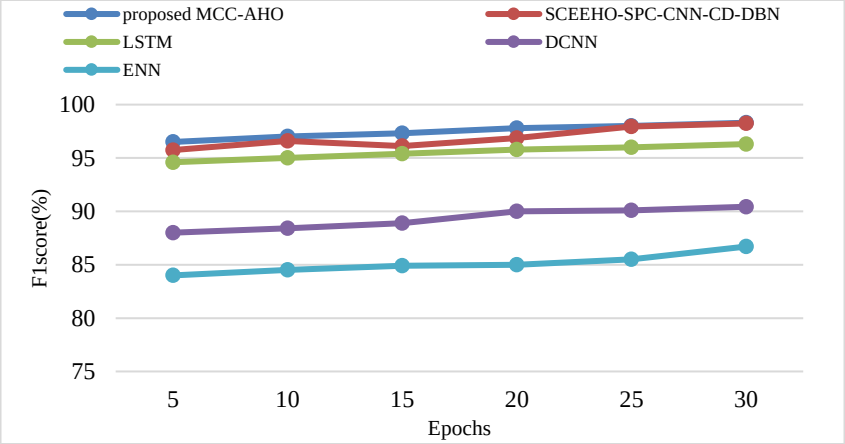

Figure 7: F1-score Result Comparison of MCC-AHO, SCEEHO-SPC-CNN-CD-DBN, LSTM, DCNN, and ENN

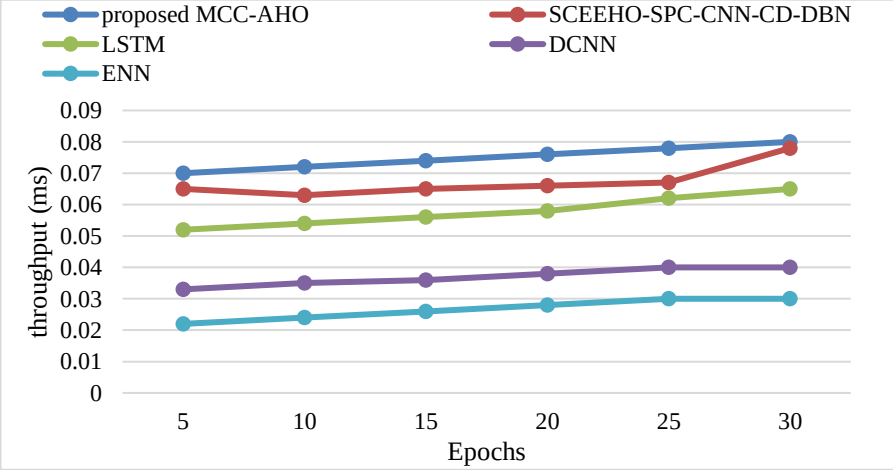

Figure 8: Throughput Result Comparison of MCC-AHO, SCEEHO-SPC-CNN-CD-DBN, LSTM, DCNN, and ENN

Figure 8 indicates the throughput of proposed MCC-AHO and existing models SCEEHO-SPC-CNN-CD-DBN, LSTM, DCNN, and ENN for the number of features in a given database. As the number of

features increases, the throughput is also maximized (Visa et al., 2011). The MCC-AHO achieves a throughput of 0.08ms compared to the other traffic classification schemes. Because existing methods are simple replicas that are unproductive for high-dimensional datasets, they are underfit. Once the best hyper-parameter configuration is determined in AHO, it will be used to fully train an MCC, which would generate a much lower error rate and thus improve the throughput of the proposed method.

5 Conclusion

This work proposed ensemble-based DL algorithms with optimization methods, also offering scalable and adaptable IDS with multi-staged binary and multi-class classifiers (Jain et al., 2018). To attain its extremely high detection accuracy, MCC-AHO relies on an ensemble technique based on both hybrid machine and DL approaches such as LSTM, DCNN, and ENN. More specifically, we furthermore used the Lambda architecture to optimize the training of binary and multi-class classifiers so that it all occurs in a single batch. Simultaneously, the low-latency speed layer uses model inferences to assess and analyse real-time IoT traffic. Further show that compared to preexisting traffic classification techniques like LSTM, DCNN, and ENN, the ensemble approach MCC-AHO achieves greater detection accuracy of 99.5%, precision of 99.4%, recall of 99.2%, f1 score of 98.3%, and throughput of 0.08ms (Miao & Zhu, 2022; Davis & Goadrich, 2006). Also, plan to use other DL techniques in the ensemble model in the future to improve detection accuracy and overall system presentation. In further planning, verify the performance of the proposed framework in a live, production IoT setting. IoT devices universally conform to essential security protocols, encompassing authentication, verification, encryption, and security enhancements. Consequently, in order to safeguard the confidentiality of data, IoT devices must employ encryption techniques to secure communications prior to transmitting them across the cloud. However, it is imperative to prioritize privacy protection during the design of IoT devices. Therefore, as a continuation of our research, researchers aim to expand this research to encompass full frameworks for privacy, security, and cyber-attacks in new IoT contexts and This is necessary as there is still a need for significant improvements in these areas.

References

[1] Ackoski, J. (2020). Man in the middle attacks. *International Journal of Science and Arts–IDEA*, *4*(8), 1-136.

[2] Ahmad, R., Alsmadi, I., Alhamdani, W., & Tawalbeh, L. A. (2022). A comprehensive deep learning benchmark for IoT IDS. *Computers & Security*, *114*, 102588. https://doi.org/10.1016/j.cose.2021.102588

[3] Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, *32*(1), e4150. https://doi.org/10.1002/ett.4150

[4] Aldeen, M., Saha, S., & Alpcan, T. (2014). Voltage stability margins and risk assessment in smart power grids. *IFAC Proceedings Volumes*, *47*(3), 8188-8195. https://doi.org/10.3182/20140824-6-ZA-1003.02102

[5] Alenezi, M., Nadeem, M., & Asif, R. (2021). SQL injection attacks countermeasures assessments. *Indonesian Journal of Electrical Engineering and Computer Science*, *21*(2), 1121-1131. http://doi.org/10.11591/ijeecs.v21.i2.pp1121-1131

[6] Aliero, M. S., Shamaki, B. I., Kalgo, B. S., & Bello, A. A. M. (2020). Web Application Firewall. *International Journal of All Research Writings*, *3*(4), 26-43.

[7] Ayesh, A. N. (2024). Enhancing Urban Living in Smart Cities Using the Internet of Things (IOT). *International Academic Journal of Science and Engineering*, *11*(1), 237-246. https://doi.org/10.9756/IAJSE/V11I1/IAJSE1127

[8] Azizyan, G., Miarnaeimi, F., Rashki, M., & Shabakhty, N. (2019). Flying Squirrel Optimizer (FSO): A novel SI-based optimization algorithm for engineering problems. *Iran. J. Optim*, *11*(2), 177-205.

[9] Azumah, S. W., Elsayed, N., Adewopo, V., Zaghloul, Z. S., & Li, C. (2021, June). A deep lstm based approach for intrusion detection iot devices network in smart home. In *2021 IEEE 7th World Forum on Internet of Things (WF-IoT)* (pp. 836-841). IEEE. https://doi.org/10.1109/WF-IoT51360.2021.9596033

[10] Bisong, E. (2019). *Building machine learning and deep learning models on Google cloud platform* (pp. 59-64). Berkeley, CA: Apress. https://doi.org/10.1007/978-1-4842-4470-8

[11] Boopathy, E. V., Niranjana, M. I., Prasath, S., Sivabalvigneshan, P., Sanjesh, R., Sanjay, S., ... & Vishal, S. (2025). Aegis Flare: Iot-Enabled Robotic Firefighter for Advanced Fire Detection and Suppression. *Archives for Technical Sciences/Arhiv za Tehnicke Nauke*, (32). https://doi.org/10.70102/afts.2025.1732.066

[12] Chadd, A. (2018). DDoS attacks: past, present and future. *Network Security*, *2018*(7), 13-15. https://doi.org/10.1016/S1353-4858(18)30069-2

[13] Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC genomics*, *21*(1), 6. https://doi.org/10.1186/s12864-019-6413-7

[14] Davis, J., & Goadrich, M. (2006, June). The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning* (pp. 233-240). https://doi.org/10.1145/1143844.1143874

[15] Elsayed, N., ElSayed, Z., & Bayoumi, M. (2023, June). IoT botnet detection using an economic deep learning model. In *2023 IEEE World AI IoT Congress (AIIoT)* (pp. 0134-0142). IEEE. https://doi.org/10.1109/AIIoT58121.2023.10174322

[16] Ghillani, D. (2022). Deep learning and artificial intelligence framework to improve the cyber security. *Authorea Preprints*.

[17] Ghimire, B., & Rawat, D. B. (2022). Recent advances on federated learning for cybersecurity and cybersecurity for federated learning for internet of things. *IEEE Internet of Things Journal*, *9*(11), 8229-8249. https://doi.org/10.1109/JIOT.2022.3150363

[18] Grochowski, E., Ronen, R., Shen, J., & Wang, H. (2004, October). Best of both latency and throughput. In *IEEE International Conference on Computer Design: VLSI in Computers and Processors, 2004. ICCD 2004. Proceedings.* (pp. 236-243). IEEE. https://doi.org/10.1109/ICCD.2004.1347928

[19] Hafez, A. I., Zawbaa, H. M., Emary, E., Mahmoud, H. A., & Hassanien, A. E. (2015, November). An innovative approach for feature selection based on chicken swarm optimization. In *2015 7th international conference of soft computing and pattern recognition (SoCPaR)* (pp. 19-24). IEEE. https://doi.org/10.1109/SOCPAR.2015.7492775

[20] Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawk's optimization: Algorithm and applications. *Future generation computer systems*, *97*, 849-872. https://doi.org/10.1016/j.future.2019.02.028

[21] Idrissi, I., Boukabous, M., Azizi, M., Moussaoui, O., & El Fadili, H. (2021). Toward a deep learning-based intrusion detection system for IoT against botnet attacks. *IAES International Journal of Artificial Intelligence*, *10*(1), 110. http://doi.org/10.11591/ijai.v10.i1.pp110-120

[22] Jain, M., Maurya, S., Rani, A., & Singh, V. (2018). Owl search algorithm: a novel nature-inspired heuristic paradigm for global optimization. *Journal of Intelligent & Fuzzy Systems*, *34*(3), 1573-1582. https://doi.org/10.3233/JIFS-169452

[23] Khan, A. R., Kashif, M., Jhaveri, R. H., Raut, R., Saba, T., & Bahaj, S. A. (2022). Deep learning for intrusion detection and security of Internet of things (IoT): current analysis, challenges, and possible solutions. *Security and communication networks*, *2022*(1), 4016073. https://doi.org/10.1155/2022/4016073

[24] Khattak, H. A., Shah, M. A., Khan, S., Ali, I., & Imran, M. (2019). Perception layer security in Internet of Things. *Future Generation Computer Systems*, *100*, 144-164. https://doi.org/10.1016/j.future.2019.04.038

[25] Kumar, T. S. (2024). Low-power design techniques for Internet of Things (IoT) devices: Current trends and future directions. *Prog. Electron. Commun. Eng.*, *1*(1), 19-25.

[26] Lahasan, B., & Samma, H. (2022). Optimized deep autoencoder model for internet of things intruder detection. *IEEE Access*, *10*, 8434-8448. https://doi.org/10.1109/ACCESS.2022.3144208

[27] Lal, B., Ravichandran, S., Kavin, R., Kumar, N. A., Bordoloi, D., & Kumar, R. G. (2023). IOT-BASED cyber security identification model through machine learning technique. *Measurement: Sensors*, *27*, 100791. https://doi.org/10.1016/j.measen.2023.100791

[28] Lata, M., & Kumar, V. (2022). IoT network security in smart homes. *Cybersecurity in smart homes: architectures, solutions and technologies*, 155-176.

[29] Li, C., Yu, X., Huang, T., & He, X. (2017). Distributed optimal consensus over resource allocation network and its application to dynamical economic dispatch. *IEEE transactions on neural networks and learning systems*, *29*(6), 2407-2418. https://doi.org/10.1109/TNNLS.2017.2691760

[30] Li, C., Yu, X., Huang, T., Chen, G., & He, X. (2015). A generalized Hopfield network for nonsmooth constrained convex optimization: Lie derivative approach. *IEEE transactions on neural networks and learning systems*, *27*(2), 308-321. https://doi.org/10.1109/TNNLS.2015.2496658

[31] Lin, K. C., & Hsieh, Y. H. (2015). Classification of medical datasets using SVMs with hybrid evolutionary algorithms based on endocrine-based particle swarm optimization and artificial bee colony algorithms. *Journal of medical systems*, *39*(10), 119. https://doi.org/10.1007/s10916-015-0306-3

[32] Malik, R., Singh, Y., Sheikh, Z. A., Anand, P., Singh, P. K., & Workneh, T. C. (2022). [Retracted] an Improved Deep Belief Network IDS on IoT-Based Network for Traffic Systems. *Journal of Advanced Transportation*, *2022*(1), 7892130. https://doi.org/10.1155/2022/7892130

[33] Martins, I., Resende, J. S., Sousa, P. R., Silva, S., Antunes, L., & Gama, J. (2022). Host-based IDS: A review and open issues of an anomaly detection system in IoT. *Future Generation Computer Systems*, *133*, 95-113. https://doi.org/10.1016/j.future.2022.03.001

[34] Mehedi, S. T., Anwar, A., Rahman, Z., Ahmed, K., & Islam, R. (2022). Dependable intrusion detection system for IoT: A deep transfer learning-based approach. *IEEE Transactions on Industrial Informatics*, *19*(1), 1006-1017. https://doi.org/10.1109/TII.2022.3164770

[35] Miao, J., & Zhu, W. (2022). Precision–recall curve (PRC) classification trees. *Evolutionary intelligence*, *15*(3), 1545-1569. https://doi.org/10.1007/s12065-021-00565-2

[36] Nakashika, T., Takiguchi, T., & Ariki, Y. (2014). Voice conversion using RNN pre-trained by recurrent temporal restricted Boltzmann machines. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *23*(3), 580-587. https://doi.org/10.1109/TASLP.2014.2379589

[37] Neto, E. C. P., Dadkhah, S., Ferreira, R., Zohourian, A., Lu, R., & Ghorbani, A. A. (2023). CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment. *Sensors*, *23*(13), 5941. https://www.unb.ca/cic/datasets/iotdataset-2023.html

[38] Novak, P., & Vacek, T. (2023). An IoT-Driven Predictive Maintenance Model for Smart Manufacturing Systems. *International Academic Journal of Innovative Research*, *10*(3), 9–15. https://doi.org/10.71086/IAJIR/V10I3/IAJIR1019

[39] Otoum, Y., Liu, D., & Nayak, A. (2022). DL-IDS: a deep learning–based intrusion detection framework for securing IoT. *Transactions on Emerging Telecommunications Technologies*, *33*(3), e3803. https://doi.org/10.1002/ett.3803
[40] Pan, J., Ye, N., Yu, H., Hong, T., Al-Rubaye, S., Mumtaz, S., ... & Chih-Lin, I. (2022). AI-driven blind signature classification for IoT connectivity: A deep learning approach. *IEEE Transactions on Wireless Communications*, *21*(8), 6033-6047.
[41] Purnama, Y., Asdlori, A., Ciptaningsih, E. M. S. S., Kraugusteeliana, K., Triayudi, A., & Rahim, R. (2024). Machine Learning for Cybersecurity: A Bibliometric Analysis from 2019 to 2023. *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.*, *15*(4), 243-258. https://doi.org/10.58346/JOWUA.2024.I4.016
[42] Rizk, Y., Hajj, N., Mitri, N., & Awad, M. (2019). Deep belief networks and cortical algorithms: A comparative study for supervised classification. *Applied computing and informatics*, *15*(2), 81-93. https://doi.org/10.1016/j.aci.2018.01.004
[43] Sarker, I. H., Khan, A. I., Abushark, Y. B., & Alsolami, F. (2023). Internet of things (iot) security intelligence: a comprehensive overview, machine learning solutions and research directions. *Mobile Networks and Applications*, *28*(1), 296-312. https://doi.org/10.1007/s11036-022-01937-3
[44] Sikarwar, T. S., Chauhan, A. S., Jain, N., & Mathur, H. (2025). Application of Predictive Analytics in IOT Data Processing. *Indian Journal of Information Sources and Services*, *15*(2), 340–348. https://doi.org/10.51983/ijiss-2025.IJISS.15.2.42
[45] Visa, S., Ramsay, B., Ralescu, A. L., & Van Der Knaap, E. (2011). Confusion matrix-based feature selection. *Maics*, *710*(1), 120-127.
[46] Wang, Z., & Wang, J. (2019). A delay-adaptive control scheme for enhancing smart grid stability and resilience. *International Journal of Electrical Power & Energy Systems*, *110*, 477-486. https://doi.org/10.1016/j.ijepes.2019.03.030
[47] Whitmore, A., Agarwal, A., & Da Xu, L. (2015). The Internet of Things—A survey of topics and trends. *Information systems frontiers*, *17*(2), 261-274. https://doi.org/10.1007/s10796-014-9489-2
[48] Zhuo, Q., Li, Q., Yan, H., & Qi, Y. (2017, November). Long short-term memory neural network for network traffic prediction. In *2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)* (pp. 1-6). IEEE. https://doi.org/10.1109/ISKE.2017.8258815

Authors Biography

Boyella Mala Konda Reddy was born in the year 1987, Andhra Pradesh, India.He has received his Master of Computer Application from SriVenkateswaraUniversity, Tirpathi in 2011 and Master of Technology in Computer science & Engineering from JNTU, Ananthapur in 2015, and Master of Business Administration from JNTU, Anantapur in 2018. He is dedicated to teaching field from the last 6 years. His research areas included Data Mining& Advanced Computer Networks & Wireless Sensor Networks & Could Areas. At present he is working as Associate Professor in Akshara Institute of Management and Technology, Tirupathi, and Andhra Pradesh, India.

Dr.A. Abdul Azeez Khan was born in the year 1982, Tamil Nadu, India. He obtained his doctorate degree in the field of computer science in the year 2018. He is having 19+ years of experience put together in industry and academia. At present he is working as Associate Professor in the Department of Computer Applications at B.S. Abdur Rahman Crescent Institute of Science and Technology, Chennai-600048. His papers are published in many International Journals & Conferences. His area of interest includes Artificial Intelligence, Machine Learning, IoT, Elearning and Knowledge Management.

Dr.K. Javubar Sathick was born in the year 1984, Tamil Nadu, India. He obtained his doctorate degree in the field of computer science in the year 2018. He is having 15+ years of experience in academics. At present he is working as Associate Professor in the Department of Computer Applications at B.S. Abdur Rahman Crescent Institute of Science and Technology, Chennai. His papers are published in many International Journals & Conferences. His area of interest includes Artificial Intelligence, Machine Learning, IoT, E-learning and Knowledge Management.

Dr.L. Arun Raj was born in the year 1985, Tamil Nadu, India. He obtained his doctorate degree in the field of computer science engineering in the year 2017. He is having 15+ years of in academics. At present he is working as Associate Professor in the Department of Computer Science and Engineering at B.S. Abdur Rahman Crescent Institute of Science and Technology, Chennai-600048. His papers are published in many International Journals & Conferences. His area of interest includes Artificial Intelligence, Machine Learning, IoT, Networking, Cyber Security.