IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 15, No. 3, June 2026, pp. 2920~2934
ISSN: 2252-8938, DOI: 10.11591/ijai.v15.i3.pp2920-2934

Class imbalance resolution in IoT networks using advanced elk herd optimization with SMOTE and iteratively fine-tuned deep BiLSTM

Srikanth Mudiyanur Sriramappa[1], Ananda Babu Jayachandra[1], Vasantha Kumara Mahadevachar[2], Ashwini Kailas[3], T. G. Keerthan Kumar[4]

[1]Department of Information Science and Engineering, Malnad College of Engineering affiliated to Visvesvaraya Technological University, Belagavi, India
[2]Department of Computer Science and Engineering, Government Engineering College affiliated to Visvesvaraya Technological University, Belagavi, India
[3]Department of Bio Medical Engineering, Sri Siddhartha Institute of Technology affiliated to Sri Siddhartha Academy of Higher Education, Tumkur, India
[4]Department of Information Science and Engineering, Siddaganga Institute of Technology affiliated to Visvesvaraya Technological University, Belagavi, India

Article Info

Article history:

Received Oct 1, 2024
Revised Feb 21, 2026
Accepted Apr 20, 2026

Keywords:

Advanced elk herd optimization
Deep BiLSTM
Information sharing
Internet of things
Intrusion detection system
SMOTE

ABSTRACT

The "internet of things (IoT)" mentions to a system in which multiple network protocols are used to link together disparate devices that are always sharing information with one another. The hope is that this study will add to the existing body of knowledge by suggesting ways to make intrusion detection systems (IDS) more effective. Training datasets for the proposed model were collected from telemetry of network (ToN)-IoT network traffic. After cleaning and normalizing the datasets, the synthetic minority over-sampling technique (SMOTE) is used to balance the datasets that are imbalanced. Optimal sampling rates are critical for resolving class imbalance, as SMOTE's efficiency is dependent on it for instances involving minority classes. Improving classification accuracy through finding appropriate sample rates for input datasets is the goal of this paper's introduction of advanced elk herd optimization (AEHO) with SMOTE. Finally, a deep bidirectional long short-term memory (deep BiLSTM) model based on deep learning is used to classify attacks. The fine-tuning technique is used during testing to update the high limits and when combined with the data balancing mechanism and AEHO-based-SMOTE, the results greatly enhance the classification techniques. Deep BiLSTM performs better than 90% in every category: classification, recall, precision, and F1-score.

Corresponding Author:

Srikanth Mudiyanur Sriramappa
Department of Information Science and Engineering
Malnad College of Engineering affiliated to Visvesvaraya Technological University
Belagavi, India
Email: srikanthise@gmail.com

1. INTRODUCTION

Intelligent technologies that enable machine-to-machine interaction are backbone of Industry 4.0, which aims to digitalize traditional industrial production and working procedures. Advanced internet of things (IoT) techniques and instruments automate all production, job processing, and operation processes in industrial automation [1]. The term "IIoT" describes a method of using the IoT for industrial automation.

Both machine-to-machine (M-M) and human-machine (H-M) communications are encompassed by this notion of industrial automation. Healthcare, manufacturing, the supply chain, and manufacturers are among the many industries that have recently shown a marked increase in their adoption of IoT technology [2]. Virtualization, automation, learning methods, artificial intelligence approaches, and the IoT are some of the most utilized tools in industry 4.0 infrastructures. The newest innovations are integrated into several areas of smart manufacturing and communiqué in industry 4.0 [3]. Info security is constantly demanding in industry 4.0 since data in all of these sectors is available on distant platforms or cloud servers, which can be targeted by intrusion assaults. The increased security risk is a result of Industry 4.0's reliance on devices and networks for communiqué and data transmission [4]. Any smart device linked to network provides a potential entry point for cybercriminals and intruders to carry out malicious operations. This could result in significant data loss and financial consequences for the organization.

No network, no matter how it is set up, can be considered secure without IDS [5]. This system is designed to identify anomalies and prevent intrusions or hosts from posing dangers to the network, providing high levels of security. Adaptability and detection of cyberattacks and threats are the main goals of intrusion detection systems (IDS). Two primary categories of intrusion detection schemes are anomaly and misuse. Incoming traffic's signatures are examined and matched against a database of known assaults by the former [6]. These technologies are very accurate in their detection, but they still can't find modified or zero-day threats. To add insult to injury, attackers are always coming up with new methods and plans, thus IDS need to be flexible to keep up. The present approach to tuning signatures might be more trustworthy, but alas, it is not [7].

By using sophisticated statistical methods to the study of network traffic patterns, anomaly-based IDS aim to circumvent the shortcomings of abuse IDSs. It can be utilized to identify suspicious occurrences as they happen. A number of statistical methods, including machine learning, are available for use in anomaly detection [8]. Anomaly IDSs are superior to signature IDSs at detecting zero-day assaults. But because they may spot unusual traffic, they can also have a high proportion of false alarms [9]. That way, they can identify malicious attempts that disrupt the normally functioning of the network. At this time, attack signatures transmitted across the IoT network cannot be detected by abused IDS [10]. Due to the lack of a universal signature for new assaults, many methods have been devised to forestall their repetition. This can be accomplished by analyzing the data acquired by network traffic using machine learning techniques. Finding security risks to IoT networks can thus be aided by machine learning [11].

The detection and monitoring capabilities of data gathered by IoT networks have been enhanced by the development and implementation of numerous machine learning models with feature reduction algorithms [12]. Despite these models' output, their reliability for IoT networks is still in question. In addition, the relative merits of various machine learning methods for constructing an adaptive IDS remain debatable. Many deep learning procedures have been tested on different datasets to find out how effective they are at detecting IoT IDS [13]. Because of how important time is to a successful IoT attack prevention system, it is crucial to think about ways to construct and train the system faster. Reducing the computational time needed by IDS can accomplish this.

When machine learning classifiers' performance is poor, researchers use the ensemble approach to build a stronger prediction model by combining their weaker classifiers [14]. Furthermore, the capacity of the machine learning detection system is database dependent; thus, it is essential to gather or create a trustworthy dataset from the IoT infrastructures setting at several levels, including actual traffic [15]. Datasets telemetry of network (ToN)-IoT, compiled from a wide variety of IoT devices, have recently undergone testing and authentication in a cyber lab to ensure their reliability for use in machine learning procedures. The following are a few of the most significant difficulties: data imbalance in the IoT environment and unknown threats, such zero-day attacks, are the most important issues discussed in this article. A major hurdle to effective intrusion detection in such situations is the issue researchers confront when dealing with diverse and imbalanced data acquired from various and distinct devices.

Within the IoT's IDS, there is another problem that needs fixing. This difficulty is related to a class of assaults that could damage the network but whose nature is unclear and cannot be predicted in advance. Like zero-day attacks, the exact time of these assaults is also impossible to foresee. As a result, conventional methods of detection grow increasingly ineffective, making them an essential subject of continuing research.

The main contribution of the research work is mentioned as follows: our research suggests a new IDS that uses deep learning procedures to monitor IoT network data for signs of infiltration. The primary objective of our method is to improve the precision of assault detection. In order to distinguish between typical and non-standard IoT network traffic, we built and tested four supervised machine learning models. Consequently, the following are the contributions of the paper:

i) Suggesting a simple way to label network traffic from IoT devices as either normal or problematic. The IoT network devices' IDS performance can be improved by using an optimization strategy to increase the synthetic minority over-sampling technique (SMOTE) sampling rate.

ii) Building and testing a model for attack detection called deep bidirectional long short-term memory (deep BiLSTM). Metrics like as part of the presentation evaluation. The built model outperformed the most recent study using the same dataset in terms of detection approach performance.

The rest of the paper is arranged as follows. Section 1 introduces IoT and IDS along with the contributions of this work. Section 2 reviews related literature. Section 3 describes proposed methodology. Section 4 presents experimental results and discussion. Lastly, section 5 concludes the paper.

2. RELATED WORKS

Kaushik *et al.* [16] studied a decision tree model with unique feature selection algorithm for detection of intrusion in the environment. The performance was improved by the designed model and nearly 27-63% of training time was minimized by classifiers. The detection accuracy was improved and the computational overhead was minimized by using the widely used discriminative features. While comparing with NSL-KDD dataset, the proposed model achieved over 99.9% of recall, precision, F1-score, and accuracy on IoT-ID20 dataset.

For the purpose of intrusion detection in IoT networks using a machine learning-based attack categorization framework, Li *et al.* [17] has conducted a comprehensive comparison of feature extraction and selection. Using classification, it examines performance parameters such as accuracy, F1-score, and runtime on the network ToN-IoT dataset, which is a heterogeneous IoT dataset. Since there are fewer characteristics to work with in feature extraction, it outperforms feature selection in terms of detection performance. Not only that, extraction is less affected by changes in the quantity of features and exhibits less feature reduction than selection. Nevertheless, when contrasted with its alternative, feature selection results in shorter model training and inference times. Additionally, with a larger feature set, there is more room to refine the selection process rather than the extraction phase. Both multiclass and binary classifications adhere to this. Finding the right intrusion detection methods for different situations is made easier with the help of the study's recommendations. Comparison and recommendations based on the ToN-IoT heterogeneous IoT dataset were previously disregarded. In sum, the study provides a comprehensive analysis of feature reduction approaches for intrusion detection in IoT networks powered by learning.

Using a deep learning system, Yaras and Dener [18] were able to analyze the collected dataset of network traffic in a large data setting and identify network threats. The research is carried out in a Google Colaboratory setting utilizing PySpark and Apache spark. The research makes use of the Scikit-learn and Keras libraries. The model is trained and tested using the 'CICIoT2023' and 'ToN_IoT' datasets. By reducing the number of features in the datasets by correlation, we can be sure that the tests will only include the most important features. The goal was to create a deep learning algorithm that combines LSTM with one-dimensional convolutional neural network (CNN). Ten different algorithms from the fields of machine learning and deep learning were tested alongside the new technique. F1 parameters, recall, accuracy, and precision were used to assess the model's performance. The research shows that using the 'CICIoT2023' dataset, the accuracy classification is 99.95% and for multiclassification it is 99.96%. A binary classification success rate of 98.75% is achieved in the 'ToN-IoT' dataset.

To improve IDS accuracy and reduce attack risk, Soltani *et al.* [19] proposed a hybrid machine learning method that uses random forests and k-nearest neighbors as supervised classifiers. Linear discriminant analysis and backward elimination also used to reduce computational costs and reduce features. When disagreements emerged between the classifiers' conclusions after training, the final verdict was backed by ISO/IEC 27001 standards. The suggested model's CICIDS 2017, NSL-KDD, besides ToN-IoT datasets within a Python programming environment. The results showed that the suggested method achieved an impressive presentation with a multi-class classification accuracy of 99.96% on the CICIDS 2017 dataset, a binary classification accuracy of 99.37% on the NSL-KDD dataset, and a multi-class classification accuracy of 99.96% on the ToN-IoT dataset. Attack success rates for each dataset are 0.05%, 0.24%, and 0%, individually, showing a considerable decrease when related to other approaches.

An innovative and efficient framework based on deep learning has been created to identify intrusions in smart agricultural systems by Kethineni and Pradeepini [20]. The first of the three levels of the design is the sensor layer, which comprises the actual installation of sensors in fields and other agricultural settings. The suggested IDS is installed in every fog node that makes up the second tier, the fog computing layer (FCL). For additional data processing, the collected information is sent to this fog layer. Data storage and end-to-end services are provided by the third tier, the cloud computing layer. To identify and categorize intrusions, the suggested model combines a CNN model with a bidirectional gated recurrent unit (BiGRU) model. The BiGRU model incorporates an attention method to discover the critical aspects that trigger the distributed denial of service (DDoS) attack. The wild horse algorithm, a meta-heuristic optimization technique inspired by nature, further improves the classification model's accuracy. Finally, data is collected

from fog nodes and storage services are provided by cloud layer, last layer. The ToN-IoT and APA-DDoS attack datasets will be used to evaluate the proposed system after it is built in the Python platform. In comparison to the current approaches, the suggested system achieves better results in terms of rate (99.02%), precision (99.89%), and F1-score (99.05%) on APA DDoS attack dataset, and on ToN-IoT dataset (99.71%).

To implement semi-supervised learning-based (IDS) without the need for attack training samples, Uddin *et al*. [21] proposed two methods: i) constructing an one class classification (OCC) model trained just on harmless network traffic and ii) training a supervised machine learning model with synthetic attack samples distributed uniformly and randomly. We put both methods to the test on ten up-to-date benchmark IDS datasets and compared their results. In terms of detecting previously unknown attacks, in particular, the results show that the OCC model trained and evaluated using real-life scenarios works far better than conventional supervised classification and other OCC based approaches. This model is based on the state-of-the-art anomaly detection procedure. An adaptive boosting (AdaBoost)-based intrusion detection framework has been proposed [22]. To bring our model to life, the chi-squared method is employed to tackle the class imbalance problem. Two datasets—UNSW-NB15 and ToN-IoT—are used to evaluate the model. F1-score, accuracy, precision, and recall are used to assess the model.

Krishnan and Shrinath [23] was used a method called localized generalized matrix learning vector quantization (LGMLVQ) to determine which features were most important for each class. Applying our model to the well-known NF-BoT-IoT dataset, it performs admirably, with an accuracy score of 99.9952%. Unseen attack detection using a shallow autoencoder with reconstruction error thresholding is another area of interest in the suggested research. The method is tested on two benchmark datasets, NF-ToN-IoT besides NF-CSE-CIC-IDS 2018, to see how well it works. The model's presentation on new samples is impressive, with an average F1-score of 95.145%, an accuracy of 93.715%, a precision of 99.955%, and a recall of 90.865%. In the setting of imbalanced data, the proposed work offers a complete solution that has proven performance in identifying both known and new attacks, making substantial addition to IoT security.

3. METHOD

In this section, the brief explanation of the projected classical method is mentioned, where Figure 1 presents the research workflow of work. Initially, the dataset that is going to validate the proposed models' analysis is described, then pre-processing techniques that is used to normalize the input dataset is mentioned. Then, the detailed description of the proposed model for attack detection is mathematically explained.

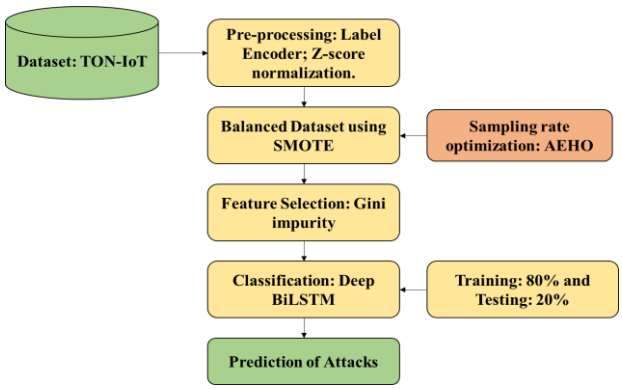

Figure 1. Workflow of the research work

3.1. Description of ToN-IoT network dataset

Data for the ToN IoT network traffic set comes from actual home-based IoT device scenarios [24]. In Canberra, at the University of New South and Information Technology (SEIT), the Australian Defense Force Academy (ADFA) is located, which is also the source of the ToN-IoT dataset network [25]. To ensure the ToN-IoT datasets, gathered from various IoT devices using IoT construction, are more trustworthy when applied and implemented using machine learning methods, they have been evaluated and genuine in a cyber lab. Nonetheless, we provide a binary categorization of typical and non-standard network traffic from IoT devices in this study. By building an effective ensemble framework of supervised machine learning models, the suggested model offers a way to enhance IDS performance.

Data from the IIoT sensors, system logs, and details of network traffic are just a few examples of the diverse types of evidence that can be gathered and analyzed using this dataset. IIoT is born from a workable network. A number of physical devices, cloud nodes, and virtual servers are all interconnected in the

ToN-IoT dataset. As part of this process, we must evaluate the efficacy and accuracy of various AI-based information security initiatives. There are cyberattacks on businesses in this dataset. A number of factors were considered in the testing, such as Windows 7 and 10, network samples, transmission details, and IoT training models, besides attacks on each. Regular, denial of service (DoS), backdoor, DDOS, man-in-the-middle (MITM), injection, ransomware, password entries are all part of the ToN-IoT dataset [26]. A comprehensive overview of the ToN-IoT dataset is provided in Table 1.

Table 1. ToN-IoT dataset imageries

Total records	Data based on events
71,401	Scanning
21,089	Cross-site scripting (XSS)
17,185	Password
79,638	Normal
33,753	DoS
45,265	Injections
7,280	Ransomware
50,811	Back-door
61,650	DDoS
105	MITM

3.2. Data pre-processing

Missing values, too many features, duplication of behavior, and other issues diminish the performance of the models used in these data samples. For the purpose of data balancing and classifier performance improvement, we employed class weights. A ranking column is included in the ToN-IoT samples for each attack subclass type, taking into account its general behavior and vulnerability state. The data was shown as a CSV file. This dataset mostly includes nine important types of attacks. The not a number (NaN) class, or records missing key information, was culled from the database.

Datasets containing categorical information will need to have it converted to numerical values before deep learning techniques may be applied. The scikit-learn package's label encoder class was used for this conversion. A numerical number was given to each attribute value. Various levels of the specimen's attributes are measured in the dataset. To properly evaluate these specimens using deep learning approaches, the taster data needs to be spoken in a consistent particular measure. This was achieved by applying the Z-score normalization technique to the dataset using the standard scaler class in the library. The dataset's observations were then normalized with a mean average of zero and a standard deviation of one. Fixed properties like dsport, srcip, dstip, and sport that were not needed were also removed, introduced regular vectors towards the "NaNclass," and set the "NaNclass" variables to zero. Each window is sent in a distinct packet so that the current datagram can identify if an attack was triggered by prior transmissions.

3.2.1. Data splitting

Partitioning the data according to the feature mapping is an important thing to keep in mind. Using randomization and stratification, the dataset was divided into training besides testing subsets with the same 80:20 class type split as the original dataset. The training data were used to build predictive models, and then we tested them on a separate set to see which intrusion detection model performed the best. Overfitting, however, can occur if a large amount of training data is chosen. This happens when classifier's performance on unseen test data suffers because it has grown overly biassed and specialized to the training data.

3.3. Solving imbalanced data

A categorical variable serves as the output in a classification task, and the goal is to estimate the class group from the input data. When one group of people has more information than another, that group is considered the majority, and when one group has less information, that group is considered the minority. When some dataset classes are overrepresented compared to others, the data is considered imbalanced [25]. Resolving imbalances in datasets is crucial when utilizing them to build a detection model. Hence, to build a final model that isn't biassed in identifying specific kinds of attacks, the trained with a suitable proportion of data from multiple classes. The imbalanced dataset issue has multiple solutions. For the purpose of oversampling minorities, the SMOTE method is employed in this work. As choosing the right sample rate is a tough task, but a frequent machine learning method is to address class imbalance procedure. Various aspects, such as dataset properties, algorithms, and problem-specific nuances, influence the appropriate sample rate, so there are no general rules for this. When training models, it is essential to maintain a balanced distribution of classes; SMOTE aims to do this. Overfitting, underfitting, or poor model performance might result from choosing the incorrect sample rate.

Before beginning work on cluster problems, researchers frequently employ SMOTE techniques to balance their datasets. This allows them to avoid rate selection for minority classes. Grid search is a method for optimizing assessment measures like area under the curve (AUC), F1-score, recall, and precision by testing out different predetermined sample rates. An improved evaluation across numerous data subsets is provided by cross-validation, which improves this procedure. Researchers often use an iterative refinement method to progressively reduce the ideal sample rate through study and testing. Knowing how machine learning algorithms react to various sample rates is also very important. Overall, selecting the optimal sample degree in SMOTE is a complex process that calls for empirical approaches, domain knowledge, and repeated exploration to find the sweet spot that works for the given dataset and problem area. To ensure data set balance, samples from minority groups will be generated using the proposed method for identifying the most accurate sampling rate. An advanced optimization method, the advanced elk herd optimization (AEHO) algorithm, is utilized by this solution.

3.3.1. Description of advanced elk herd optimization

First, the constraints discovered during lengthy experimentation with benchmark functions are discussed, followed by an explanation of the baseline variation of elk herd optimization (EHO) metaheuristics. Then, an upgraded version of EHO is released, with the intention of further maximizing the performance level of original EHO. The EHO metaheuristics are considered to be among the most current procedures because EHO was proposed in early 2024 [27]. It is a part of the population-based strategies that draw inspiration from nature, namely the mating and breeding behaviors of elk.

There are two primary phases to these activities: calving. In the first stage, the herd is divided into families of varying sizes. The bulls' struggle for supremacy directs the process of separation, with the strongest bulls mating with multiple females to establish families. In the second stage, the dominant male and associated females in each family give birth to new calves. A single control variable, Br, representing the first bull ratio in the populace, characterizes the baseline form of EHO.

The initial population, an elk herd with bulls and harems, is generated before each metaheuristic execution. In (1) shows a matrix representing the herd EH, with dimensions n×N, where N is the herd size.

$$EH = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_n^1 \\ x_1^2 & x_2^2 & \cdots & x_n^2 \\ \vdots & \vdots & \cdots & \vdots \\ x_1^N & x_2^N & \cdots & x_n^N \end{bmatrix} \quad (1)$$

Every single elk x^j is produced rendering to (2).

$$x_i^j = lb_i + (ub_i - lb_i) \times U(0,1) \quad (2)$$

In which ub besides lb are the bounds of the key domain, respectively. The elks in the group are then ranked from best to worst according to their fitness levels.

During the mating season, the male rate Br is used to establish the family unit. It is possible to first ascertain the total number of families by $B = |Br \times N|$. The fitness values of the males in the population are used for selection purposes. The strongest males in a herd fight for dominance, leading to families. The top B elks in terms of fitness are called bulls, representing this elite group.

Therefore, in order to start families, the bulls in B fight. Using their fitness scores relative to the cumulative fitness, the bulls in B are assigned females using the roulette wheel technique. It is specifically assigned a probability for each guy x_j in B. p_j, which is strong-minded by its complete fitness marked by $f(x^j)$, excluded from consideration due to the total fitness of all males, as seen in (3).

$$p_j = \frac{f(x^j)}{\sum_{k=1}^{B} f(x^k)} \quad (3)$$

In phase, of every family, labeled $x_i^j(t+1)$, is fashioned based chiefly on from the paternal bull x^{h_j} besides elk, denoted as $x_i^j(t)$. In cases where the calf $x_i(t+1)$ uses the same index (i) as the family's patrilineal bull and is created using (4).

$$x_i^j(t+1) = x_i^j(t) + a.\left(x_i^k(t) - x_i^j(t)\right) \quad (4)$$

The rate of passing on qualities from the arbitrarily picked elk in the populace can be found using (4), where α can be any value between 0 and 1 $x^k(t)$. Greater values of α lead to a greater likelihood of arbitrary augments diversification.

Another possibility is that the mother and calf share the same index. Then $x_i(t+1)$ is going to be derived from mother x^j and the father x^{h_j} based on the (5).

$$x_i^j(t+1) = x_i^j(t) + \beta\left(x_i^{h_j}(t) - x_i^j(t)\right) + \left(x_i^r(t) - x_i^j(t)\right) \quad (5)$$

Where $x_i^j(t+1)$ in the $t+1$-th restatement, h_j is the bull of the j-th harem besides r remains the index of a randomly chosen bull; this is because, in the wild, there's a certain chance that the bulls in the herd if the bull didn't protect her properly. Finally, the variables γ and β, which can take on values between 0 and 2, are used to randomly choose the ratio of traits passed down from previously produced calves.

Bulls, females, and calves from all the different families are gathered together in the next stage. In this system, the elks are ranked according to their fitness level, with the best elks being kept for the next generation. Improved elk herd optimization algorithm: a lot of testing with benchmark functions has revealed that the baseline EHO, which is an innovative method, might be even better. In order to improve the search space coverage, this publication recommends including a learning technique (quasi-reflexive learning (QRL)) during the procedure's initialization stage [28]. Regarding each variable $j(X_j)$, a quasi-reflexive-opposite limit (X_j^{qr}) will be rendering to (6).

$$X_j^{qr} = rnd\left(\frac{lb_j + ub_j}{2}, x_j\right) \quad (6)$$

Where rnd agrees to the random sum from $\left[\frac{lb_j+ub_j}{2}, x_j\right]$. When starting an EHO algorithm, the first step is to generate NP/2 solutions using the QRL technique, all without adding complexity to the algorithm in terms of fitness function evaluations (FFEs). This is a typical way to assess the complexity of metaheuristic algorithms, since computing the fitness function is the costliest process that occurs during algorithm execution. Here is the recommended method for initialization. Algorithm 1 explains the stages of the proposed model.

Algorithm 1. QRL initialization practice
Stage 1: construct starting populace Pinit having NP/2 individuals through traditional
EHO initialization scheme provided in (2)
Stage 2: construct QRL populace Pqr upon Pinit by employing the (6)
Stage 3: construct final initial populace P as union of Pinit and Pqr (P ∪ Pqr)
Stage 4: compute the fitness score for every solution in P
Stage 5: arrange all solutions belonging to P with respect to their fitness scores

Starting from the initialization stage and continuing throughout the metaheuristic's execution, the worst solution is removed and substituted with the QRL inverse of the best distinct (directed best method) for each iteration. The complication of the baseline technique calculated in FFEs is not increased by this introduced adjustment because the fitness scores are not reviewed. The EHO also had another modification that was prompted by a genetic algorithm [29]. The search center on the top solutions discovered thus far as the iterations progress and the procedure starts to converge. At the end of the max_iter/2 rounds—where max_iter is the maximum number of iterations—the algorithm is sped up by swapping out the solution with the second-score for a new individual. This new individual is a cross of the two best solutions obtained after put on an operator, where pc is the parameter of the individual and the probability of crossover per gene is 0.1. This update speeds up the algorithm by reinforcing the exploitation. Once again, this change doesn't add any extra work for the FFEs because it doesn't add any further fitness value calculations. As a result, the modified EHO and the baseline EHO are equally complex. The modified version of EHO is now known as accelerated guided Algorithm 2 shows the pseudocode for this optimization iteration, where t is a variable.

Algorithm 2. AEHO pseudocode

```
Synthesize initial population P with QRL as explained in Algorithm 1
t = 0
while (t < max_iter) do
Sort solutions within P regarding their fitness scores
for each solution X in P do
Employ EHO search to conduct optimization
Synthesize fresh solution as QRL opposite of the current best
Replace the worst solution within P with this fresh solution
if (t > max_iter 2) then
Produce new individual as the hybrid of the best pair of individuals by applying
```

```
uniform crossover operator having pc = 0.1
Replace the second-worst solution with this novel hybrid individual
end if
end for
t = t + 1
end while
return the best solution found within P
```

3.4. Feature selection

Feature selection is a method for decreasing a dataset's dimensionality or the amount of features it contains. It achieves this by sifting through the raw data in search of useful and pertinent information. It takes these features and uses them to generate a smaller set of features. Conventional wisdom holds that each class in a training set is equally important; nevertheless, this ignores the possibility of an uneven distribution of training data. This method understood the importance of the characteristics in the balanced data by using a weight adjustment mechanism in random forest. The method identified the efficacy of a split in dividing the total samples of a particular class in a particular node after examining the Gini impurity.

3.5. Classification with deep BiLSTM

Even while recurrent neural networks (RNNs) are great at retrieving historical data, they can't handle exploding gradients very well, therefore they're not very useful for long-term tasks. The solution to this problem was to use deep BiLSTM, an extension of BiLSTM, a specialized LSTM version. By collecting persistent patterns in spatial-temporal information, this architecture is great at analyzing sequential spatial features in the immediate vicinity. Figure 2 shows the deep BiLSTM model's architecture in detail, highlighting its capacity to thoroughly analyze and categorize the ToN-IoT dataset's multi-class assaults.

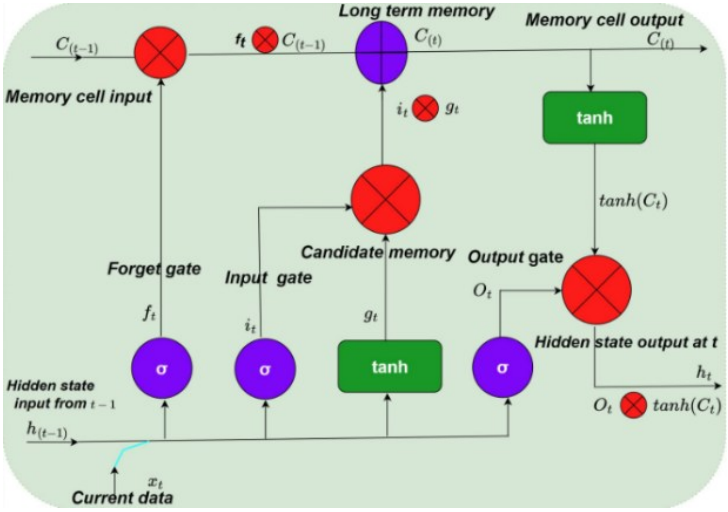

Figure 2. Architectures of LSTM [30]

The three gates—gate, the forget gate, and the output gate—are what make LSTM the basis of the deep BiLSTM model. In (7) is a detailed description of an LSTM unit.

$$i_t = \sigma(\Upsilon_i \oplus x_t + W_i \oplus h_{t-1} + b_i) \quad (7)$$

This is where Y besides W stand for the weight matrix, b for the bias term, x_t for the input data at a given time, $\oplus$ for multiplication procedure, and σ for the function. (t), besides h_{t-1} represents the result of the LSTM unit that came before it. The input gate plays a pivotal role in determining which data points from the previous unit require editing. The forget gate is formulated in (8).

$$f_t = \sigma\left(\Upsilon_f . x_t + W_f . h_{t-1} + b_c\right) \quad (8)$$

Where f_t stands for gate, which is in charge of calculating the importance of data and erasing previous data. The candidate cell state is given in (9) and the cell state update is defined in (10).

$$\tilde{c}_t = tamh(\Upsilon_c . x_t + W_c . h_{t-1} + b_c) \quad (9)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (10)$$

Where the c_t public of the applicant is unwavering through the operation of the tangent function, as demonstrated in (10). Then, the state is assessed and providing in equation, where $\odot$ characterizes multiplication. The output gate is computed in (11) and the hidden state output is given in (12).

$$g_t = \sigma\left(\Upsilon_g x_t + W_g h_{t-1} + b_g\right) \quad (11)$$

$$h_t = g_t \odot \tanh\left(c_t\right) \quad (12)$$

Given the output of the LSTM unit, denoted as ht, as shown in (12), the output gate gt is computed using (11). The baseline LSTM model uses historical data exclusively to forecast human actions in the present. The risk of missing important details when data is only analyzed in one way is obvious. Figure 3 demonstrations that the BiLSTM consists of two LSTM layers that can function in both directions. The following is the formulation of v_t, the output in the deep BiLSTM, is given in (13) [31].

$$v_t = \left[\overleftarrow{h_t} \overrightarrow{h_t}\right] \quad (13)$$

The forward besides backward consequences LSTM units is characterized by the symbols $\overleftarrow{h_t}$ and $\overrightarrow{h_t}$. In order to get the output v_t, these two LSTM units work together. The core idea behind RNN is that when data is fed into the neural network, the whole dataset is not provided at once, but rather in a sequential fashion, essentially adding the temporal variable together. If there is a set of input values, the process may start by feeding the network the beginning value and obtaining the appropriate output. Then, the following input is fed alongside the previous output to get the following outcome.

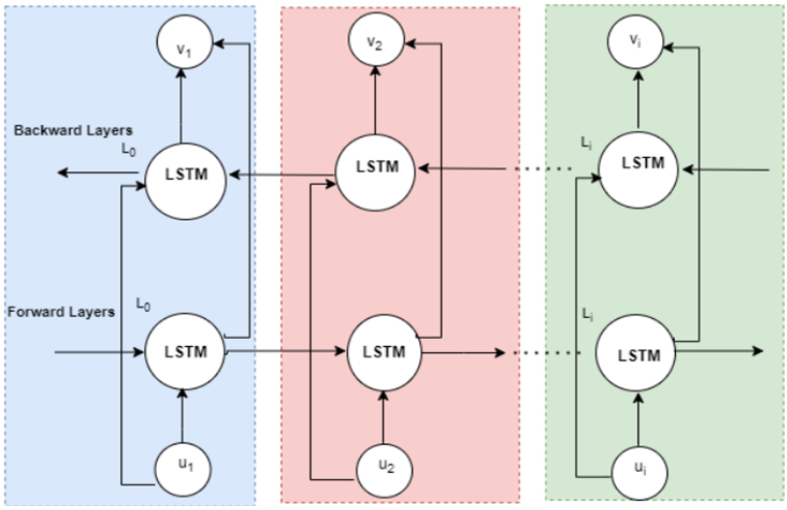

Figure 3. Structure diagram of deep BiLSTM

4. RESULTS AND DISCUSSION

Windows 11 operating system having 32 GB of DDR4 RAM powered by a Core i7-12700 processor is used for simulation environment. The desktop computer's hard disc capacity is 2 terabytes. The NVIDIA RTX 3080 is the GPU supporting the graphics card.

An indicator of how well a machine learning classification system is doing is the confusion matrix [32]. The model's performance can be evaluated in various classification outcomes using this method, which is based on the association among real classes besides the classes foretold by the classical (Figure 4). The x-axis shows the predictable outcomes, besides the y-axis expressions the observed outcomes. In a true positive (TP) situation, the actual and forecast classifications are in agreement; in a true negative (TN)

situation, the actual besides predicted classifications are at odds with one another. It is called a false positive (FP) when the actual classification is wrong but the anticipated classification is right. When the anticipated classification is wrong but the classification is right, it's called a false negative (FN).

4.1. Evaluation metrics

In order to determine how well and accurately the suggested model identifies attacks, the evaluation process is of the highest importance. A confusion matrix is used to measure the suggested classical in terms of accuracy, recall, precision, besides F1-score. Researches can objectively evaluate the system's performance on these important metrics by using this matrix. Results from accurately and efficiently detecting and identifying assaults using the suggested methodology were encouraging. On binary attacks, Figure 5 shows the suggested BiLSTM's confusion matrix.

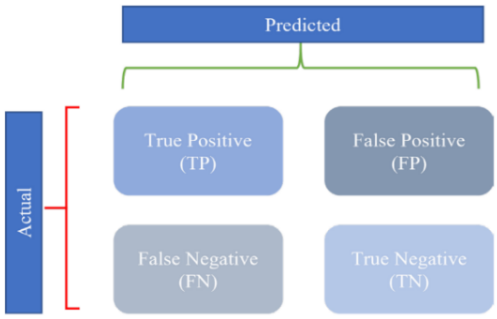

Figure 4. Confusion matrix

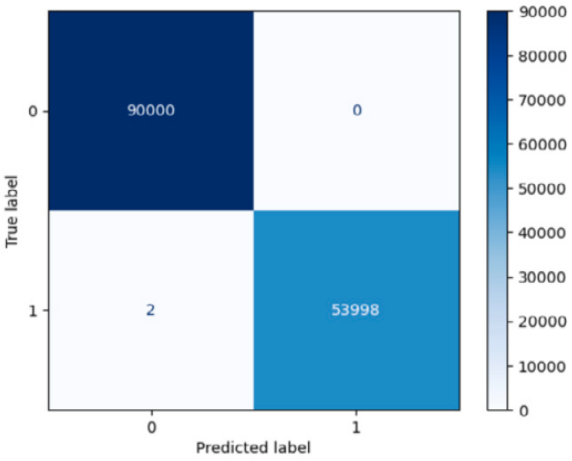

Figure 5. Confusion matrix of the proposed BiLSTM

4.2. Validation analysis of proposed deep learning model for binary class

The effectiveness of the proposed model is validated with existing techniques such as Markov random field (MRF) [16], 1D-CNN and LSTM [18], BiGRU [20], and AdaBoost [22]. The existing techniques use various datasets and therefore the research work implements these models on dataset. The results are averaged in Figures 6 and 7. Figure 6 demonstrations the comparative investigation of the suggested perfect. The MRF [16] analysis yielded a classifier accuracy of 94.98 and a corresponding F1-score of 89.02. Next, the F1-score was 90.41 besides the classifier accuracy was 96.24 according to AdaBoost [22]. The classifier accuracy for the CNN and LSTM [18] was then 97.52 and 93.57, respectively, for the F1-score. Then, the corresponding BiGRU [20] classifier accuracy was 98.48~95.82. The deep BiLSTM classifier accuracy was then reported as 99.06 with a corresponding F1-score of 97.48.

In Figure 7 signifies the visual representation of different models in terms of several metrics. In the investigation of MRF [16] technique precision as 93.52 and recall of 91.57 correspondingly. Then the

AdaBoost [22] technique precision as 94.05 and recall of 93.48 correspondingly. Then the CNN and LSTM [18] technique precision as 96.87 and recall 95.06 correspondingly. Then the BiGRU [20] technique precision as 97.80 and recall of 98.24 correspondingly. Then the deep BiLSTM technique precision as 98.07 and recall of 99.08 correspondingly.

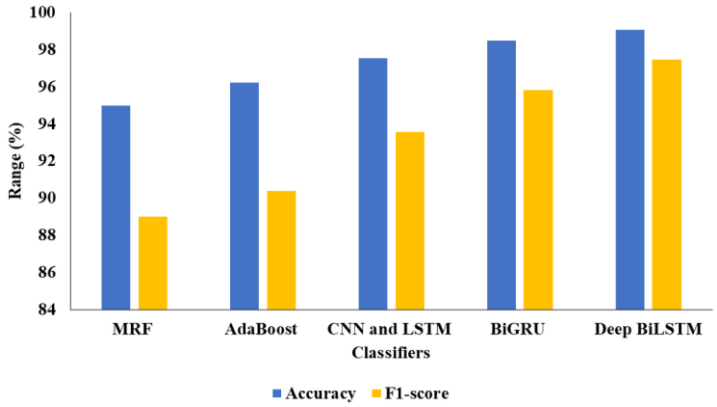

Figure 6. Comparative analysis of projected model

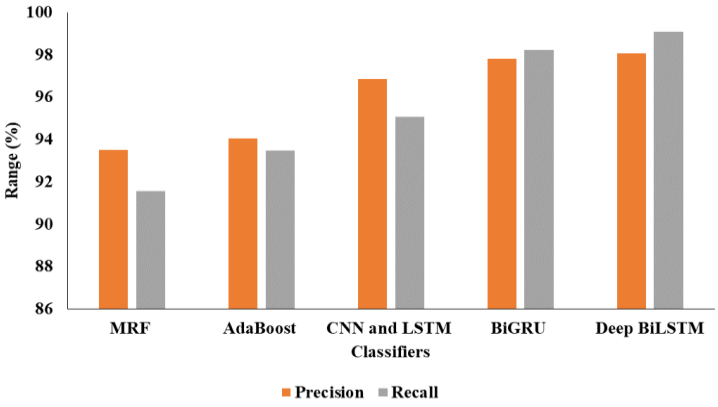

Figure 7. Different models in terms of several metrics

4.3. Validation investigation of proposed model on multi-class data

The presentation of the projected model is tested with dissimilar types of attacks with existing classifies in terms of accuracy, that is shown in Figure 8. Figure 8 represent the analysis of projected perfect on attacks. In the analysis of normal attack type, the MRF [16] technique attained the accuracy rate as 90.54, AdaBoost [22] technique reached as 93.26, CNN-LSTM [18] rate as 96.47, BiGRU [20] reached as 98.24, and deep BiLSTM reached as 99.17 correspondingly. Then the DoS attack type, the MRF [16] technique attained the accuracy rate as 91.56, AdaBoost [22] technique reached as 93.27, CNN-LSTM [18] rate as 95.69, BiGRU [20] reached as 94.57, and deep BiLSTM reached as 98.07 correspondingly. Then the back-door attack type, the MRF [16] technique attained the accuracy rate as 89.16, AdaBoost [22] technique reached as 95.06, BiGRU [20] reached as 96.75, and deep BiLSTM reached as 97.35 correspondingly. Then the DDoS attack type, the MRF [16] technique attained the accuracy rate as 92.63, AdaBoost [22] technique reached as 94.09, CNN-LSTM [18] rate as 95.07, BiGRU [20] reached as 95.69, and deep BiLSTM reached as 96.57 correspondingly. Then the MITM attack type, the MRF [16] technique attained the accuracy rate as 87.06, AdaBoost [22] technique reached as 88.53, CNN-LSTM [18] rate as 91.02, BiGRU [20] reached as 93.46, and deep BiLSTM reached as 94.27 correspondingly. Then the injections attack type, the MRF [16] technique attained the accuracy rate as 88.59, AdaBoost [22] technique reached as 89.43, CNN-LSTM [18] rate as 90.47, BiGRU [20] reached as 91.36, and deep BiLSTM reached as 93.56 correspondingly. Then the ransomware attack type, the MRF [16] technique attained the accuracy rate as 86.40, AdaBoost [22] reached

as 89.06, BiGRU [20] reached as 90.48, and deep BiLSTM reached as 91.24 correspondingly. Then the scanning attack type, the MRF [16] technique attained the accuracy rate as 78.25, AdaBoost [22] reached as 83.19, BiGRU [20] reached as 86.43, and deep BiLSTM reached as 88.48 correspondingly. Then the XSS attack type, the MRF [16] technique attained the accuracy rate as 80.29, AdaBoost [22] achieved 83.44, BiGRU [20] reached as 89.05, and deep BiLSTM reached as 90.25 correspondingly. Then the password records attack type, the MRF [16] technique attained the accuracy rate as 81.95, AdaBoost [22] reached as 85.34, BiGRU [20] reached as 87.16, and deep BiLSTM reached as 89.15 correspondingly.

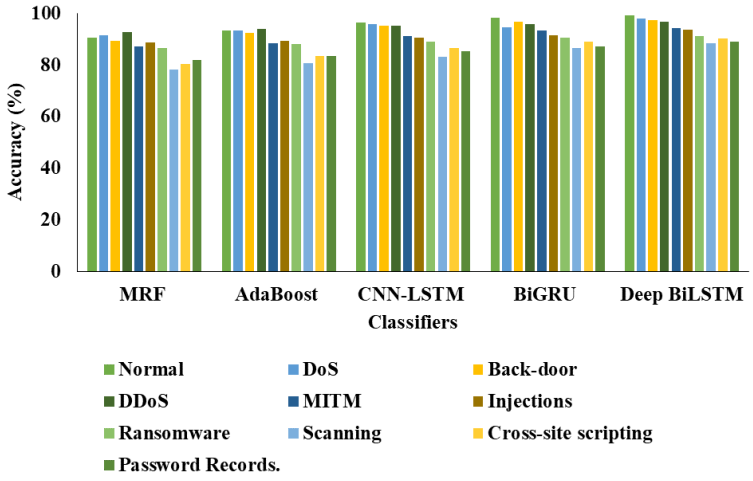

Figure 8. Analysis of anticipated perfect on multi-class attacks

4.4. Performance analysis of proposed optimization

In test the effectiveness of projected AEHO model, this research work tests its performance in terms of different metrics with existing optimization models such as grey wolf optimization (GWO), besides also the butterfly optimization algorithm (BOA), wild horse optimization algorithm (WHOA), and standard EHO. The graphical description is mentioned in Figure 9. Figure 9 characterize that the optimization model performance of projected with different compared optimization techniques. In the analysis of GWO optimization accuracy of 84.18, precision rate of 88.52, recall measures as 91.57, and then F1-score of 86.02 correspondingly. Then the BOA optimization accuracy of 86.34, precision rate of 89.05, recall measures as 92.48, and then F1-score of 87.41 correspondingly. Then the WHOA optimization accuracy of 87.52, precision rate of 90.87, recall measures as 94.06, and then F1-score of 89.57 correspondingly. Then the EHO optimization accuracy of 90.48, precision rate of 92.80, recall measures as 96.24, and then F1-score of 92.82 correspondingly. Then the AEHO Optimization accuracy of 93.06, precision rate of 94.07, recall measures as 97.08, and then F1-score of 94.48 correspondingly.

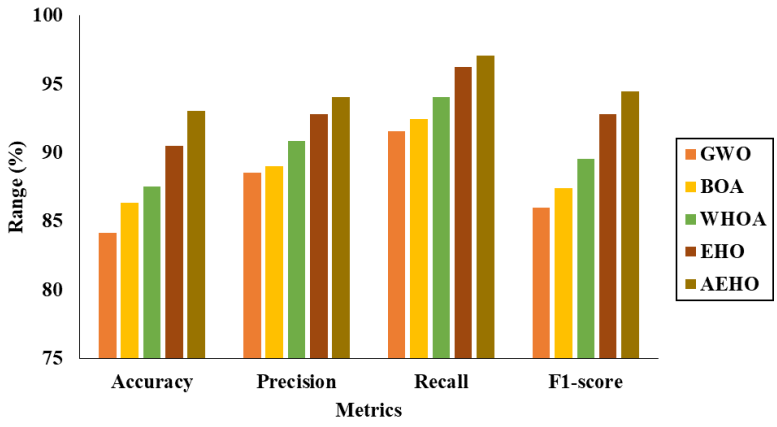

Figure 9. Visual representation of projected optimization

5. CONCLUSION

The suggested model improves the IDS's detection capability by using deep BiLSTM for attack detection. To address the issue of biassed classification caused by over-fitting and under-fitting in imbalanced datasets, we implemented SMOTE method. The AEHO model determines the ideal sample rate for the SMOTE. When it comes to detecting cyberattacks in a network like this, several machine learning besides proposed deep learning-based intrusion finding systems is employed. Using ToN-IoT, a newly created IoT dataset, for both classification, thorough testing was conducted. There should be more emphasis on the generalizability of the classification algorithms, even though this research is trying to improve their performance for a specific feature set. The algorithm is expected to be trained on more diverse datasets in the future to enable it to detect a wider variety of cyberattacks. On top of that, a decision-making unit has been integrated into the system so that it can respond appropriately to detected threats. To further reduce the detection models' training duration, other feature selection methods will be considered.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Srikanth Mudiyanur Sriramappa	✓	✓			✓		✓		✓	✓	✓		✓	
Ananda Babu Jayachandı			✓	✓		✓		✓		✓		✓	✓	
Vasantha Kumara Mahadevachar	✓		✓			✓	✓			✓	✓	✓	✓	
Ashwini Kailas		✓		✓	✓			✓		✓	✓			
T. G. Keerthan Kumar			✓			✓	✓		✓	✓			✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

INFORMED CONSENT

We have obtained informed consent from all individuals included in this study.

ETHICAL APPROVAL

No such human or animals are involved in the project.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author, [SMS], upon reasonable request.

REFERENCES

[1] B. Xu, L. Sun, X. Mao, R. Ding, and C. Liu, "IoT intrusion detection system based on machine learning," *Electronics*, vol. 12, no. 20, Oct. 2023, doi: 10.3390/electronics12204289.
[2] M. Catillo, A. Pecchia, and U. Villano, "A deep learning method for lightweight and cross-device IoT botnet detection," *Applied Sciences*, vol. 13, no. 2, Jan. 2023, doi: 10.3390/app13020837.

[3] F. Færøy, M. Yamin, A. Shukla, and B. Katt, "Automatic verification and execution of cyber attack on IoT devices," *Sensors*, vol. 23, no. 2, Jan. 2023, doi: 10.3390/s23020733.
[4] Z. Zhao *et al.*, "CMD: co-analyzed IoT malware detection and forensics via network and hardware domains," *IEEE Transactions on Mobile Computing*, vol. 23, no. 5, pp. 5589–5603, May 2024, doi: 10.1109/TMC.2023.3311012.
[5] V. S. Anusuya, S. Baswaraju, A. Thirumalraj, and A. Nedumaran, "Securing the MANET by detecting the intrusions using CSO and XGBoost model," in *Intelligent Systems and Industrial Internet of Things for Sustainable Development*, Boca Raton: Chapman and Hall/CRC, 2024, pp. 219–234, doi: 10.1201/9781032642789-11.
[6] A. Gaurav, B. B. Gupta, and P. K. Panigrahi, "A comprehensive survey on machine learning approaches for malware detection in IoT-based enterprise information system," *Enterprise Information Systems*, vol. 17, no. 3, Mar. 2023, doi: 10.1080/17517575.2021.2023764.
[7] M. El Bouazzati, R. Tessier, P. Tanguy, and G. Gogniat, "A lightweight intrusion detection system against IoT memory corruption attacks," in *2023 26th International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS)*, May 2023, pp. 118–123, doi: 10.1109/DDECS57882.2023.10139718.
[8] Y. Al Sawafi, A. Touzene, and R. Hedjam, "Hybrid deep learning-based intrusion detection system for RPL IoT networks," *Journal of Sensor and Actuator Networks*, vol. 12, no. 2, Mar. 2023, doi: 10.3390/jsan12020021.
[9] S. Pundir, M. S. Obaidat, M. Wazid, A. K. Das, D. P. Singh, and J. J. P. C. Rodrigues, "MADP-IIME: malware attack detection protocol in IoT-enabled industrial multimedia environment using machine learning approach," *Multimedia Systems*, vol. 29, no. 3, pp. 1785–1797, Jun. 2023, doi: 10.1007/s00530-020-00743-9.
[10] V. Srinivasan, V. H. Raj, A. Thirumalraj, and K. Nagarajan, "Detection of data imbalance in MANET network based on ADSY-AEAMBi-LSTM with DBO feature selection," *Journal of Autonomous Intelligence*, vol. 7, no. 4, Jan. 2024, doi: 10.32629/jai.v7i4.1094.
[11] P. R. K. Varma, R. R. Sathiya, and M. Vanitha, "Enhanced Elman spike neural network based intrusion attack detection in software defined internet of things network," *Concurrency and Computation: Practice and Experience*, vol. 35, no. 2, Jan. 2023, doi: 10.1002/cpe.7503.
[12] A. Omotosho, Y. Qendah, and C. Hammer, "IDS-MA: intrusion detection system for IoT MQTT attacks using centralized and federated learning," in *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, Jun. 2023, pp. 678–688, doi: 10.1109/COMPSAC57700.2023.00093.
[13] S. I. Popoola, A. L. Imoize, M. Hammoudeh, B. Adebisi, O. Jogunola, and A. M. Aibinu, "Federated deep learning for intrusion detection in consumer-centric internet of things," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 1610–1622, Feb. 2024, doi: 10.1109/TCE.2023.3347170.
[14] N. Jeffrey, Q. Tan, and J. R. Villar, "Intrusion detection and prevention in industrial internet of things: a study," in *International Joint Conference 16th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2023) 14th International Conference on EUropean Transnational Education (ICEUTE 2023)*, 2023, pp. 37–48, doi: 10.1007/978-3-031-42519-6_4.
[15] J. Bhayo, S. A. Shah, S. Hameed, A. Ahmed, J. Nasir, and D. Draheim, "Towards a machine learning-based framework for DDOS attack detection in software-defined IoT (SD-IoT) networks," *Engineering Applications of Artificial Intelligence*, vol. 123, Aug. 2023, doi: 10.1016/j.engappai.2023.106432.
[16] S. Kaushik *et al.*, "Robust machine learning based Intrusion detection system using simple statistical techniques in feature selection," *Scientific Reports*, vol. 15, Feb. 2025, doi: 10.1038/s41598-025-88286-9.
[17] J. Li, M. S. Othman, H. Chen, and L. M. Yusuf, "Optimizing IoT intrusion detection system: feature selection versus feature extraction in machine learning," *Journal of Big Data*, vol. 11, no. 1, Feb. 2024, doi: 10.1186/s40537-024-00892-y.
[18] S. Yaras and M. Dener, "IoT-based intrusion detection system using new hybrid deep learning algorithm," *Electronics*, vol. 13, no. 6, Mar. 2024, doi: 10.3390/electronics13061053.
[19] N. Soltani, A. M. Rahmani, M. Bohlouli, and M. Hosseinzadeh, "Robust intrusion detection for network communication on the internet of things: a hybrid machine learning approach," *Cluster Computing*, vol. 27, no. 7, pp. 9975–9991, Oct. 2024, doi: 10.1007/s10586-024-04483-7.
[20] K. Kethineni and G. Pradeepini, "Intrusion detection in internet of things-based smart farming using hybrid deep learning framework," *Cluster Computing*, vol. 27, no. 2, pp. 1719–1732, Apr. 2024, doi: 10.1007/s10586-023-04052-4.
[21] M. A. Uddin, S. Aryal, M. R. Bouadjenek, M. Al-Hawawreh, and M. A. Talukder, "usfAD based effective unknown attack detection focused IDS framework," *Scientific Reports*, vol. 14, no. 1, Nov. 2024, doi: 10.1038/s41598-024-80021-0.
[22] S. Amaouche, C. Hazman, A. Guezzaz, S. Benkirane, and M. Azrour, "Intrusion detection framework using AdaBoost algorithm and chi-squared technique," in *Blockchain and Machine Learning for IoT Security*, New York: Chapman and Hall/CRC, 2023, pp. 92–111, doi: 10.1201/9781003438779-6.
[23] D. Krishnan and P. Shrinath, "Robust botnet detection approach for known and unknown attacks in IoT networks using stacked multi-classifier and adaptive thresholding," *Arabian Journal for Science and Engineering*, vol. 49, no. 9, pp. 12561–12577, Sep. 2024, doi: 10.1007/s13369-024-08742-y.
[24] T. M. Booij, I. Chiscop, E. Meeuwissen, N. Moustafa, and F. T. H. den Hartog, "ToN_IoT: the role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion data sets," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 485–496, Jan. 2022, doi: 10.1109/JIOT.2021.3085194.
[25] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "TON_IoT telemetry dataset: a new generation dataset of IoT and IIoT for data-driven intrusion detection systems," *IEEE Access*, vol. 8, pp. 165130–165150, 2020, doi: 10.1109/ACCESS.2020.3022862.
[26] B. Gunapriya, A. Thirumalraj, V. S. Anusuya, B. P. Kavin, and G. H. Seng, "A smart innovative pre-trained model-based QDM for weed detection in soybean fields," in *Advanced Intelligence Systems and Innovation in Entrepreneurship*, 2024, pp. 262–285, doi: 10.4018/979-8-3693-0790-8.ch015.
[27] M. A. Al-Betar, M. A. Awadallah, M. S. Braik, S. Makhadmeh, and I. A. Doush, "Elk herd optimizer: a novel nature-inspired metaheuristic algorithm," *Artificial Intelligence Review*, vol. 57, no. 3, Feb. 2024, doi: 10.1007/s10462-023-10680-4.
[28] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Quasi-oppositional differential evolution," in *2007 IEEE Congress on Evolutionary Computation*, Sep. 2007, pp. 2229–2236, doi: 10.1109/CEC.2007.4424748.
[29] S. Mirjalili, "Genetic algorithm," in *Evolutionary Algorithms and Neural Networks*, Cham, Switzerland: Springer, 2019, pp. 43–55, doi: 10.1007/978-3-319-93025-1_4.
[30] S. Mekruksavanich and A. Jitpattanakul, "LSTM networks using smartphone data for sensor-based human activity recognition in smart homes," *Sensors*, vol. 21, no. 5, Feb. 2021, doi: 10.3390/s21051636.
[31] A. Radman and S. A. Suandi, "BiLSTM regression model for face sketch synthesis using sequential patterns," *Neural Computing and Applications*, vol. 33, no. 19, pp. 12689–12702, Oct. 2021, doi: 10.1007/s00521-021-05916-9.

[32] D. Chicco, N. Tötsch, and G. Jurman, "The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation," *BioData Mining*, vol. 14, no. 1, Feb. 2021, doi: 10.1186/s13040-021-00244-z.

BIOGRAPHIES OF AUTHORS

Srikanth Mudiyanur Sriramappa is pursuing Ph.D. at Malnad College of Engineering, Hassan affiliated to Visvesvaraya Technological University, Belagavi, India. His area of interest is IoT, machine learning, and deep neural networks. He can be contacted at email: srikanthise@gmail.com.

Ananda Babu Jayachandra holds a doctoral degree in Computer Science and Engineering from Visvesvaraya Technological University, Belagavi. He is working as professor in Information Science and Engineering at Malnad College of Engineering, Hassan. He has guided 6 students towards their doctoral degree and has published 40+ articles in reputed international conferences and SCI/Scopus/WoS journals. He can be contacted at email: abj@mcehassan.ac.in.

Vasantha Kumara Mahadevachar holds a doctoral degree in Computer Science and Engineering from Visvesvaraya Technological University, Belagavi. He is working as assistant professor in Computer Science and Engineering at Government College of Engineering, Hassan. His research interests span cross computer vision, industrial IoT, and machine learning applications. He has published more than 10 publications journals or conferences of high quality. He can be contacted at email: cmn.vasanth@gmail.com.

Ashwini Kailas holds doctoral degree from Visvesvraya Technological University from Belagavi, India. Her research interests are deep neural network, IoT, machine learning, and industrial IoT. She is working at Sri Siddartha Institute of Technology, Tumakuru. She can be contacted at email: ashwinik@ssit.edu.in.

T. G. Keerthan Kumar works at Siddaganga Institute of Technology, Tumakuru affiliated to Visvesvraya Technological University from Belagavi, India. His research interests are deep neural network, IoT, and machine learning. He can be contacted at email: keerthankumartg@sit.ac.in.