Alexandria Engineering Journal 103 (2024) 88–97

Contents lists available at ScienceDirect

Alexandria Engineering Journal

journal homepage: www.elsevier.com/locate/aej

A novel approach of botnet detection using hybrid deep learning for enhancing security in IoT networks

Shamshair Ali [a], Rubina Ghazal [a], Nauman Qadeer [b], Oumaima Saidani [c], Fatimah Alhayan [c], Anum Masood [d,*], Rabia Saleem [e], Muhammad Attique Khan [f,*], Deepak Gupta [g,h]

[a] University Institute of Information Technology, PMAS Arid Agriculture University Rawalpindi, Rawalpindi 46300, Pakistan
[b] Department of Computer Science, Federal Urdu University of Arts, Science &Technology, Islamabad 45570, Pakistan
[c] Department of Information Systems, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, P.O.Box 84428, Riyadh 11671, Saudi Arabia
[d] Department of Physics, Norwegian University of Science and Technology, Trondheim NO-7491, Norway
[e] Department of Information Technology, Government College University, Faisalabad 38000, Pakistan
[f] Department of Computer Science and Mathematics, Lebanese American University, Lebanon
[g] Department of Computer Science & Engineering, Maharaja Agrasen Institute of Technology, Delhi, India
[h] Chitkara University, Punjab, India

ARTICLE INFO

Keywords:
Cyber security
IoT Botnets
Unknown cyber-attacks
IoT networks
Cyber-physical systems
Zero-day vulnerability
Hybrid deep learning

ABSTRACT

In an era dominated by the Internet of Things (IoT), protecting interconnected devices from botnets has become essential. This study introduces an innovative hybrid deep learning model that synergizes LSTM Auto Encoders and Multilayer Perceptrons in detecting botnets in IoTs. The fusion of these technologies facilitates the analysis of sequential data and pattern recognition, enabling the model to detect intricate botnet activities within IoT networks. The proposed model's performance was carefully evaluated on two large IoT traffic datasets, N-BAIoT2018 and UNSW-NB15, where it demonstrated exceptional accuracy of 99.77 % and 99.67 % respectively for botnet detection. These results not only demonstrate the model's superior performance over existing botnet detection systems but also highlight its potential as a robust solution for IoT network security.

1. Introduction

Online services and the internet have become essential parts of daily life in recent years [1]. This technological advancement has changed the way we work, engage, and socialize, with the exponential growth of Internet of Things (IoTs) [2]. IoT is a network comprising numerous devices, interconnected networks, sensors, and systems with the potential to communicate and exchange data over the internet. These devices can interact and work together autonomously because they are embedded with a variety of technologies that allow them to gather, transmit, and receive information [3]. This interconnectedness and ability to share data in real-time form the foundation of IoT's transformative potential across various industries and everyday life [4].

According to Annual Report by [5], around 14.7 billion devices from IoT domain to be seamlessly integrated into digital landscape of the Internet by 2023. Projections for 2030 estimate an incredible 25.4 billion active IoT gadgets. This exponential increase underlines the widespread influence of IoT throughout industries and sectors [6]. However, this reputation has a dark side, as cyber adversaries target IoT for malicious sports. IoT's interconnected architecture offers adequate possibilities for cyber attackers to leverage these devices as potential conduits for orchestrated cyberattacks. Malicious actors have drastically employed botnets to launch cyber offensives towards IoT-enabled systems [7].

A botnet, controlled by some kind of a botmaster, operates through Command and Control also termed as (C&C) channel [8]. Its lifecycle involves stages like injection, connection, malicious activities, and maintenance. Malware is injected into IoT devices, which then download additional malware from a network database [9]. Bots receive instructions from C&C server for executing activities. The botmaster updates malware to maintain control [10]. In addition to traditional botnets, there exists a particularly formidable threat known as zero-day

* Corresponding authors.
E-mail addresses: rubinaghazal@uaar.edu.pk (R. Ghazal), nauman.qadeer@fuuast.edu.pk (N. Qadeer), ocsaidani@pnu.edu.sa (O. Saidani), fnalhayan@pnu.edu.sa (F. Alhayan), anum.masood@ntnu.no (A. Masood), rabiasaleem@gcuf.edu.pk (R. Saleem), attique.khan@ieee.org (M.A. Khan).

https://doi.org/10.1016/j.aej.2024.05.113
Received 19 March 2024; Received in revised form 22 April 2024; Accepted 29 May 2024
Available online 12 June 2024

botnets. These malicious networks leverage vulnerabilities in software or hardware that are previously unknown to the vendor, giving them an advantage in evading detection and mitigation efforts [11].

The initial stage of a zero-day botnet attack often involves exploiting these undisclosed or zero-day vulnerabilities, allowing the attacker to infiltrate a network of compromised devices. Once compromised, these devices proceed to download additional malware from a network database through various communication protocols. A zero-day vulnerability is some newly discovered flaw in a system without a known fix. Immediate vendor response is crucial to mitigate potential exploits [12]. Once these vulnerabilities are made public, the risk of exploitation rises, incentivizing attackers seeking financial gain or access to sensitive data. Delayed fixes increase the risks of being exploited by zero-day attacks.

Many large scale tech giants including oracle and Microsoft's had already faced severe consequences by delayed patch releases for newly reported zero-day vulnerabilities. In 2012, Oracle came to know of Java-related zero-days, but their patch came too late, leading to exploitation [13]. Similarly, Microsoft's delayed fix for a Word vulnerability resulted in political and financial damage [14]. This highlights the critical importance for IT vendors to speed up patch releases for effective product protection. Fig. 1 illustrates how the zero-day vulnerability get discovered by hackers and how they take advantage of it to exploit the system.

Based on past experiences, signature-based, rule-based, and machine learning algorithms had been a hit in figuring out previously recognized and absolutely useful attacks that monitor discriminatory patterns [16–18]. These are typically known as signatures or fingerprints. Some assaults are genuinely risky as they are able to alternate memory [19], exchange bytes over the network[20], interactions between components [21], communication buses [22], active threads and opened files [23], packets routing [24]. From the past few years, there has been observed a big advancement in cybersecurity by deep learning models. These models are now widely applied in various domains to detect cyber-attacks and deal with them intelligently in different cyber physical systems.

This shift reflects the rapid progress in the field [25]. Many AI-based cybersecurity systems are proposed recently. The proposed works in [22–24] used bidirectional LSTM based intelligent role based access control. Similarly, the authors in [25,26] have used machine learning approaches for mitigation of cyber-attacks. Such AI-based systems have outperformed traditional methods like SEIM Solutions, IPS, UTM, firewalls, and antiviruses [26]. They excel in accuracy, speed of response and adaptability to cyberattacks thus reducing investigation time and enhancing network security. Over 60 % of attacks are detected only after damage; hence, automated security strategies are crucial to effectively combat evolving threats [15].

The major contribution of this research include:

- Innovative hybrid deep learning model combining LSTM Autoencoders with MLP for IoT botnet detection.
- Demonstrated higher detection accuracy for IoT botnets and zero-day attacks compared to existing models.
- Enhanced the model's robustness against zero-day attacks by learning unusual patterns without prior specific knowledge.
- Validated the model's effectiveness and efficiency through extensive testing with real-world IoT datasets.

The manuscript is organized as follows. The current section highlights the importance of artificial intelligence in cyber security and IoTs for detecting different kinds of attacks. The next section discusses the literature review about similar research works and highlights their limitations. Section 3 discusses models and datasets used in the experiment and finally presents the proposed hybrid deep learning model for detecting IoT botnets. Then results of this study are presented in section 4th of manuscript followed by discussion in Section 5 along with limitations and future works.

2. Literature review

Alzaqebah et. al in [27] deployed Grey Wolf Optimization (GWO) technique, which is bio-inspired technique resulting in better overall detection rates and performance of previously used IDS in identifying both malicious and normal data packets over the network. The key enhancements focus on the smart initialization step, that combines both wrapper and filter techniques to help in incorporating the informative

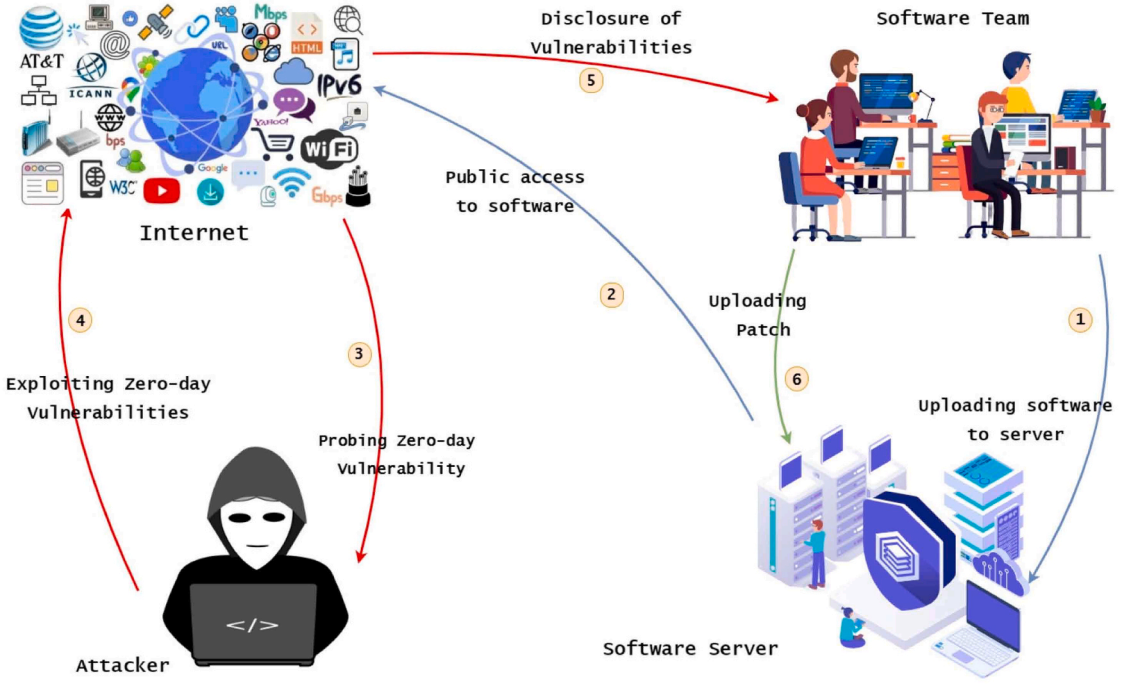

Fig. 1. Zero-day attack detection scenario in IT industry [15].

Table 1
Comparative analysis of different techniques to detect zero-days.

Paper	Algorithm	Dataset	Results	Limitations
[27]	Modified Grey Wolf Optimization (MGWO)	UNSW-NB15	Accuracy 80.93 % f1 score 78.08 % FPR 28.08 %	High false positive rate (28.08 %) Used only 17 features out of 49
[28]	BiLSTM	IOT-23	Accuracy 84.8 % Precision 36.1 % Recall 32.5 %	Low precision and recall high false alarms
[29]	KNN and ELM based hybrid model	NSL-KDD	Detection rate for known attacks 84.29 % Detection rate for zero-days 77.18 %	Performance decreased on zero-days
[31]	Ensemble modal based on NB	CICIDS-2017	binary class accuracy 89.92 % multiclass accuracy 88.96 %	Only 17 features were selected out of 80 that can lead to inconsistent results
[30]	Graph based technique	CICIDS-2018 & Custom Dataset	Accuracy 91.33 % for binary class 88.98 % for multiclass	High false positives Performance decreased on multiclass
[31]	Autoencoder	KDD-CUP	Accuracy 80 % Precision 80 % recall 78 %	Used only 19 features out of 41 Used very old dataset
[32]	DBN	TON-IOT	Accuracy 86 % Precision 78 %	High false alarms
[33]	ANN	Custom dataset	Accuracy 77.51 % Error rate 24.29 %	Higher error rate
[34]	Gated Recurrent Unit (GRU)	UNSW-NB15	accuracy 87.93 % TPR 95.79 % FPR 26 % FAR 12.06 %	High false positives
[35]	Autoencoder-GAN	private dataset	F1-score 96.85 % Recall 95.65 % Accuracy 98.24 %	Only 8 features were used in the dataset
[36]	Voting	UNSW-NB 15	Binary Acc 99.0 % Multiclass Acc 82.4 % Multiclass recall 64.9 %	Performance decreased on multiclass
[37]	HITD	CERT	Accuracy: 87 % Precision 77 % f1 score 84 %	Low precision high false alarms
[38]	BLSTM	CIC-IDS 2017	Precision 88.86 % Recall 91.95 % F1-score 90.03 %	High Misclassification rate
[39]	CNN-LSTM	InSDN	Accuracy 96.32 %	Trained on a small dataset. Only a few hundred attack samples were used in training
[40]	LSTM-Autoencoder	InSDN	Accuracy 90.5 % Precision 93 % recall 93 % F1-score 93 %	Performance decreased on zero-days

features in initial steps. Furthermore, they employed the modified GWO to tweak the settings of the Extreme Learning Machine (ELM), a high-speed categorization approach. Using the UNSWNB-15 dataset, the suggested approach was compared against other meta-heuristic algorithms. The purpose of the research was to detect generic assaults in network traffic because they are the extremely widespread type of attacks in dataset. By reducing the crossover error rate to less than 30 % they were able to achieve some better results in the accuracy, F1-score, and G-mean metrics, with 81 %, 78 %, and 84 %, respectively.

Another study by [28] contributed in analyzing and applying a strategy to protect the IoT devices in health care sector against botnet assaults. The system was able to achieve the goal by observing the communication network traffic continuously and categorizing it as normal or attack data packets. The proposed approach analyses IoT traffic from the health sector using BiLSTM. The dataset used for this research was IoT-23. The suggested method's performance was measured using a new evaluation metric known as attack prediction accuracy designed especially for this work by combining the traditional performance metrics. It achieved the maximum prediction accuracy of 84.8 % in distinguishing between normal and malicious communications.

Another study by Latah and Toker [29] suggested a hybrid classification approach to increase the system's overall accuracy. Authors applied kNN technique for the first level, after that they applied the Extreme Learning Machine (ELM) followed by Hierarchical Extreme Learning Machine (HELM) for the subsequent levels. In assessment with traditional supervised and unsupervised algorithms trained and deployed over NSL-KDD dataset, comparative analysis revealed that their system achieved an accuracy of (84.29 %), with a 77.18 % capacity to identify new attacks(zero-day). Table 1 shows the comprehensive comparison of different AI based techniques that have been deployed to detect well-known and zero-day attacks.

Kumar and Sinha [30] discussed the traditional zero-day attacks detection techniques and illustrated that to defend against these assaults, previous systems leverage ml or an anomaly-based strategies. These approaches omit numerous characteristics, such as the frequency or occurrence of some specific type of byte streams. Dealing with the attacks that occur rarely and result in less traffic is a challenge while working with simple neural networks due to the factor that they expect a large volume of dataset in training the model. Authors introduced a unique and robust attack detection model that addresses the difficulties raised by utilizing the traditional approaches for detecting zero-days.

The suggested solution was divided into two steps: (a) generating the signature of different attacks and (b) performance evaluation of model using test data and live data. During the training, the proposed model assesses performance using the signatures it has generated and then it is evaluated over test data. The proposed solution performed better on real-time data, with the detection accuracy 91 % for detecting binary

and 90.35 % in case of multi-class. The performance of this model against the benchmark dataset CICIDS18 was promising with 91.62 % accuracy for classifying binary-class. Although it outperformed the traditional approach still it was not able to overcome the false negatives and leave the system vulnerable to some security threats.

To categorize the data packets as normal or attacks from the signature databases, [31] employed different ML and DL algorithms. The correlation approach was utilized to uncover relevant network characteristics with a high percentage association among the classes and features of the dataset. They selected nine features from the dataset to use in their models. The category characteristics were converted into numerical features using a one-shot encoding technique. The system's authenticity validated by the benchmark KDD Cup database. The proposed study's results were evaluated using statistical analysis methods. To categorize the regular and attack packets, binary and multi-class classifications were used.

Malik, Singh [32] showed that the state-of-the-art methods failed to detect zero-day threats due to their unseen nature. That is why an intelligent method is required to identify these kinds of attacks efficiently. The authors of this study proposed a DBN based solution for detecting the attacks from the network data which they named intrusion detection engine. DBNclassifier was tested over a chunk of TON-IOT which consist of Weather data. The results showed that the suggested technique surpasses the other previously used state-of-the-art approaches by achieving 86.3 % average accuracy.

Khatun, Chowdhury [33] demonstrated that the Deep Learning has already proved its exceptional capabilities when dealing with heterogeneous data of varying sizes. They presented a DL technique for identifying suspicious nodes in IoT settings by applying Artificial Neural Network. Their study makes a two-fold contribution. First, it distinguishes between normal and suspicious traffic patterns of IoT devices, and then it offers a technique for identifying suspicious nodes. They achieved the accuracy of 77.51 % which can be improved by making a hybrid model combining with some other deep learning model.

In a recent study Koroniotis, Moustafa [34] presented an Intelligent Satellite based Deep Learning Network for Forensic data $(\mathrm{INSAT}-\mathrm{DLNF})$ framework for the identification and tracking of abnormal activities that were affecting the smart satellite networks. They trained a hybrid model based on LSTM-RNN and GRU and made a comparative analysis by comparing their results with their results to five different supervised learning techniques along with two unsupervised learning techniques. The results show that the combination of multiple deep learning algorithms may be applied effectively for cyber-attack detection. In comparison to conventional forensic methods that were not able to detect zero-day attack surfaces and routes properly it can result in a more adaptable and robust system.

Another study by Al-Obaidi et.al [35] utilized different ml techniques over UNSW-NB 15 dataset and illustrated a comparison of multiple models out which the Voting outperformed with and accuracy of 99.0 % for binary classification but for the multiclass their results dropped to 82 % as shown in the Table 1. Similarly in a recent study [36] authors used a hybrid dl model based on AE and GAN, experiment was conducted on their private dataset of over 1.6 M size. They achieved 98.24 % accuracy with recall of 95 % but they use only 8 features in their experiment from their data. Similarly, some other experiments are also listed in the table below along with their results and other details.

3. Proposed methodology

This section provides detailed analysis about used deep learning models along with their learning parameters and evaluation metrics.

3.1. Deep learning models

The following three dl models form the basis for proposed hybrid deep learning model. All these models are being trained over the same hyper parameters to maintain consistency for the fair comparison. For the 1st dataset n-BaIoT 2018 input parameters used were 115 that were the total features of the dataset and for the 2nd dataset UNSW-NB 15, 49 input parameters were utilized out of total 49 features of the dataset.

3.1.1. Autoencoders

Autoencoders are the type of neural networks particularly designed for unsupervised learning. They are made up of an encoder part followed by a decoder, and their primary goal is to figure out how to compress the incoming data. Tasks like dimensionality reduction, feature extraction, or reconstruction may benefit from this.

Encoder Function: Let's denote the encoder as the function $f(x)$, where x is the input data. The encoder part of the model maps the given data to a lower-dimensional latent space (Space vector) representation:

$$z = f(x) \tag{1}$$

Decoder Function: The decoder function is $g(z)$ where z is the compressed representation. The decoder maps back the compressed representation to the original input space:

$$(\widehat{x} = g(z)) \tag{2}$$

Objective Function: One of the objectives of an autoencoder is typically to minimalize the error that usually occurred at the time of reconstruction, which is the difference between the input data (x) and the output of the decoder $(\widehat{x})$, it is also used to detect the attacks from this difference. The objective function also sometimes termed as loss function can be stated as

$$(L(x, \widehat{x}) = ||x - \widehat{x}|\,|^2) \tag{3}$$

where $(||\cdot||)$ denotes a suitable norm (e.g., Euclidean norm).

3.1.2. LSTM networks

LSTM is an RNN (recurrent neural network) based neural network architecture that is designed to handle sequences of data along with backtracking or feedback. They are particularly effective at capturing long-range dependencies throughout the sequential data, making them well-suited for different tasks including natural language processing, time series analysis, and more.

LSTMs have specific gating mechanisms that control the flow of information through the network. For each time step(t), an LSTM unit can be mathematically represented as follows:

Input Gate (i):

$$i_t = \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \tag{4}$$

$$\text{Forget Gate(f)} : f_t = \sigma\left(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}\right) \tag{5}$$

$$\text{Cell Gate(g)} : g_t = \tanh\left(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}\right) \tag{6}$$

$$\text{Output Gate(o)} : o_t = \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \tag{7}$$

$$\text{Cell State(C)} : C_t = f_t \cdot C_{t-1} + i_t \cdot g_t) \tag{8}$$

$$\text{Hidden State(h)} : h_t = o_t \cdot \tanh(C_t) \tag{9}$$

3.1.3. MLP (multilayer perceptron)

It is a feedforward neural network including one or more hidden layers between input and output layers. These types of neural networks are capable of approximating complex, nonlinear functions and are widely used for tasks like classification, regression, and more.

Hidden Layer Activation: Let's denote the hidden layer's activation function as (σ). For every neuron in hidden layer, the activation is calculated as:

$$a_j = \sigma\left(\sum_{i=1}^{n} w_{ij}x_i + b_j\right) \tag{10}$$

where (w_{ij}) is the weight connecting input (i) to neuron (j), (x_i) is the input, with (b_j) as the bias value.

Output Layer Activation: Similarly, for the output layer, the activation is calculated as:

$$a_k = \sigma\left(\sum_{j=1}^{m} w'_{kj}h_j + b'_k\right) \tag{11}$$

where (w'_{kj}) termed as the weight connecting the neuron (j) in hidden layer to output (k), (h_j) is output, and (b'_k) is the bias.

Loss Function: The loss function for a typical classification task using SoftMax activation in the output layer can be defined as:

$$L(y, \hat{y}) = -\sum_{i=1}^{N} y_i \log(\hat{y}_i) \tag{12}$$

where (y) is the true label distribution and $(\hat{y})$ as predicted distribution.

3.2. Proposed LAE-MLP model

The proposed LAE-MLP (LSTM-Autoencoder-Multilayer Perceptron) is a hybrid model that combines the strengths of LSTM-Autoencoders, for sequence analysis and feature extraction, and MLP for classification. LSTM-Autoencoder is known for its ability to capture temporal patterns and sequential dependencies in data. By incorporating this component into hybrid model leverages its strength in learning hierarchical representations and encoding important sequential features. The proposed model consists of two main parts, LSTM-encoder, and MLP-decoder. Fig. 2 illustrates the proposed hybrid LAE-MLP model.

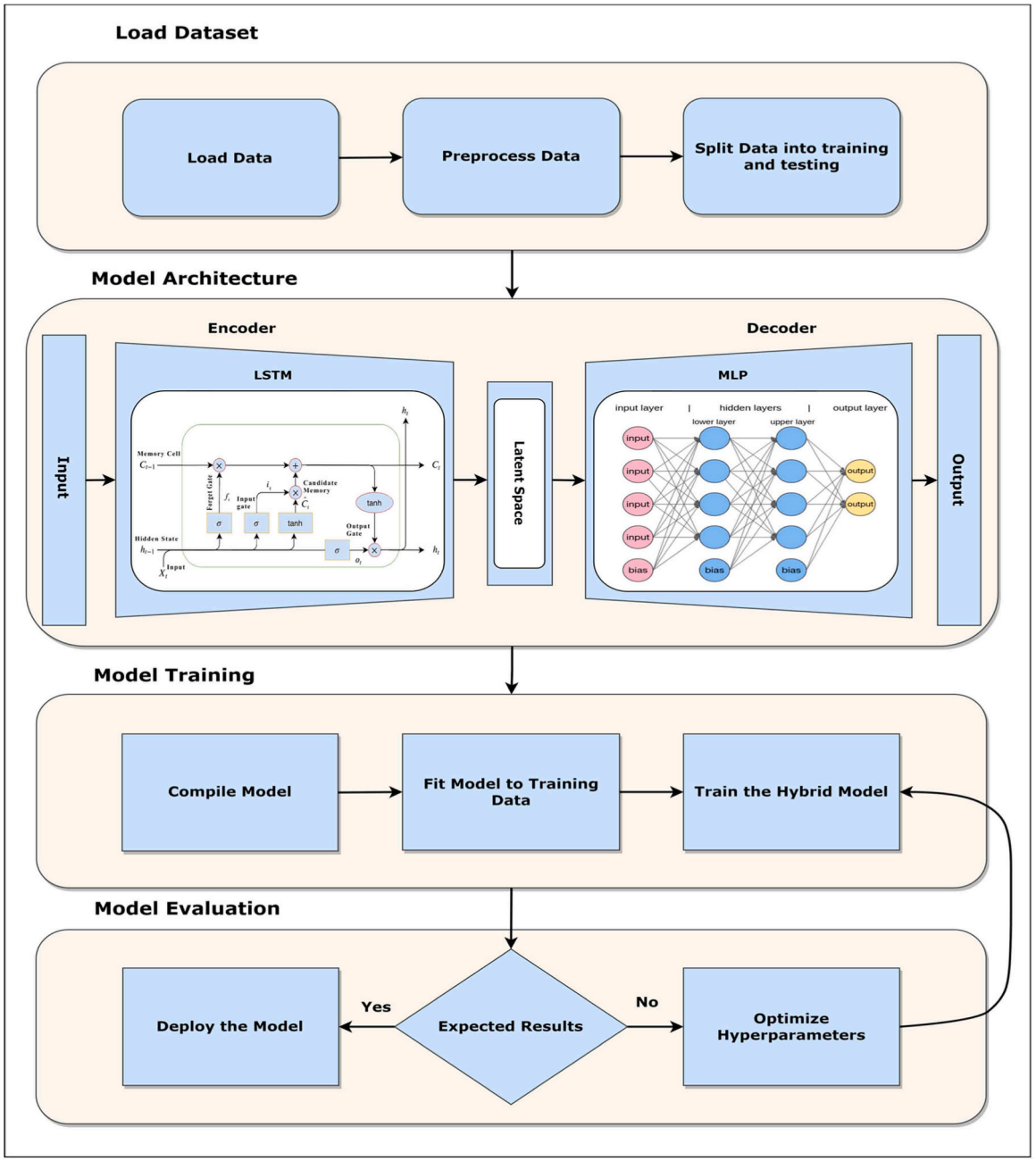

Fig. 2. Proposed Hybrid LAE-MLP Model.

The innovation of our approach lies in the unique integration of LSTM Autoencoders and Multilayer Perceptron (MLP) tailored specifically for IoT botnet detection. While each component—LSTM and MLP—is well-established, their synergistic combination to address the distinct and complex challenges of IoT security, particularly in detecting sophisticated botnet activities and zero-day attacks, constitutes a significant advancement. This model not only combines these techniques in a novel configuration but also optimizes them for the unique operational constraints and data characteristics of IoT environments, enhancing detection capabilities and efficiency.

MLP-LAE Anomaly Detection Algorithm

Zero-day attacks often exhibit complex and non-linear relationships in the data. MLP-decoder is designed to tackle the dynamic features, providing a valuable match to the LSTM-encoder's sequential pattern learning. This combination enhances the model's ability to discern novel attack patterns that may not be well-represented by linear models. MLPs are adept at learning intricate patterns and can contribute to improving the model's overall performance. This hybrid model can effectively capture intricate patterns in IoT intrusion detection data while also reducing data dimensionality. Integrating these techniques into intrusion detection systems for IoT can significantly enhance the capability to detect anomalies and safeguard IoT devices and networks from potential

---

Input: Training dataset with labeled instances $(X_{\text{train}}, y_{\text{train}})$ Testing dataset for evaluation X_{test}

Output: Anomaly predictions for the testing dataset.

Initialize LSTM Autoencoder (LAE) and MLP with specified architecture.

Training LSTM:

For each epoch t and sequence (X_i, y_i) :

Forward pass through LSTM: $\mathbf{h}^{(t)} = \text{LSTM}(\mathbf{X}^{(t)})$.

Compute loss: $\mathcal{L}_{\text{LSTM}} = \text{Loss}(\mathbf{y}, \hat{\mathbf{y}})$.

Backpropagation to compute gradients: $\frac{\partial \mathcal{L}_{\text{LSTM}}}{\partial \mathbf{W}_{\text{LSTM}}}, \frac{\partial \mathcal{L}_{\text{LSTM}}}{\partial \mathbf{b}_{\text{LSTM}}}$.

Update LSTM weights: $\mathbf{W}_{\text{LSTM}} \leftarrow \mathbf{W}_{\text{LSTM}} - \alpha_{\text{LSTM}} \frac{\partial \mathcal{L}_{\text{LSTM}}}{\partial \mathbf{W}_{\text{LSTM}}}$,

$\mathbf{b}_{\text{LSTM}} \leftarrow \mathbf{b}_{\text{LSTM}} - \alpha_{\text{LSTM}} \frac{\partial \mathcal{L}_{\text{LSTM}}}{\partial \mathbf{b}_{\text{LSTM}}}$.

Encode Sequences with Autoencoder (LAE):

For each sequence $\mathbf{X}_i$ in X_{train} :

Forward pass through LSTM: $\mathbf{h}^{(t)} = \text{LSTM}(\mathbf{X}^{(t)})$.

Feed sequence into Autoencoder (LAE): $\mathbf{z}_{\text{LAE}} = \text{LAE}(\mathbf{h}^{(t)})$.

Train MLP:

For each epoch t and instance (x_i, y_i) :

Forward pass through MLP: $\mathbf{z}^{(l+1)} = \sigma(\mathbf{W}^{(l)}\mathbf{z}^{(l)} + \mathbf{b}^{(l)})$ for $l = 1, 2, \ldots, L-1$.

Compute loss: $\mathcal{L}_{\text{MLP}} = \text{Loss}(\mathbf{y}, \hat{\mathbf{y}})$.

Backpropagation to compute gradients: $\frac{\partial \mathcal{L}_{\text{MLP}}}{\partial \mathbf{W}^{(l)}}, \frac{\partial \mathcal{L}_{\text{MLP}}}{\partial \mathbf{b}^{(l)}}$.

Update MLP weights: $\mathbf{W}^{(l)} \leftarrow \mathbf{W}^{(l)} - \alpha \frac{\partial \mathcal{L}_{\text{MLP}}}{\partial \mathbf{W}^{(l)}}$,

$\mathbf{b}^{(l)} \leftarrow \mathbf{b}^{(l)} - \alpha \frac{\partial \mathcal{L}_{\text{MLP}}}{\partial \mathbf{b}^{(l)}}$.

Inference Phase:

For each instance x_i in X_{test}:

Forward pass through LSTM: $\mathbf{h}^{(t)} = \text{LSTM}(\mathbf{X}^{(t)})$.

Encode sequence with Autoencoder (LAE): $\mathbf{z}_{\text{LAE}} = \text{LAE}(\mathbf{h}^{(t)})$.

Forward pass through MLP: $\mathbf{z}^{(l+1)} = \sigma(\mathbf{W}^{(l)}\mathbf{z}^{(l)} + \mathbf{b}^{(l)})$ for $l = 1, 2, \ldots, L-1$.

Table 2
Proposed Model results for binary classification on n-BAIoT2018 dataset.

Metric	Score
Accuracy	99.77
Precision	99.92
Recall	99.83

threats.

Initializing and training LSTM: The first step involves initializing an LSTM network along with predefined hyperparameters including epochs, hidden layers, units, training (input) parameters along with fixed batch size and optimal learning rate. The LSTM is trained over training dataset $((X_{\text{train}}, y_{\text{train}}))$. During training, for each epoch (t)and sequence $((X_i, y_i))$, a forward pass is performed through the LSTM, generating hidden states:

$$\mathbf{h}^{(\mathbf{t})} = \text{LSTM}\left(\mathbf{X}^{(\mathbf{t})}\right) \tag{13}$$

The loss $\mathscr{L}_{\text{LT}}$ is calculated and used in backpropagation to compute gradients. The LSTM weights (W_{LT}) and biases (b_{LT}) are updated accordingly:

$$\mathscr{L}_{\text{LT}} = \text{Loss}(y, \hat{y}) \tag{14}$$

Encode Sequences with Autoencoder: With the trained LSTM, each sequence (X_i) in (X_{train}) is passed through the LSTM to v generate hidden states$(h^{(t)})$. These states are then fed into an Autoencoder (LAE), which encodes the sequences into a lower-dimensional space, producing encoded sequences:

$$\boldsymbol{z}_{\boldsymbol{LAE}} = \text{LAE}\left(h^{(t)}\right) \tag{15}$$

This step is essential for capturing the salient features of the sequences, which is crucial for effective anomaly detection.

Training MLP: Next, a Multi-Layer Perceptron (MLP) is initialized and trained using the encoded sequences(zLE). The training process involves passing each instance through the MLP layers, where each layer computes

$$z^{(l+1)} = \sigma\left(W^{(l)}z^{(l)} + b^{(l)}\right) \text{for}(l = \quad 1, \quad 2, \quad \ldots, L-1 \quad) \tag{16}$$

The loss of model is calculated using:

$$M\mathscr{L}P = \text{Loss}(y, \hat{y}) \tag{17}$$

and backpropagation is used to update the MLP weights.

$$W^{(l)} \leftarrow W^{(l)} - \alpha \frac{\partial \mathscr{L}_{\boldsymbol{MLP}}}{\partial \boldsymbol{W}^{(l)}} \tag{18}$$

and biased value through the below equation:

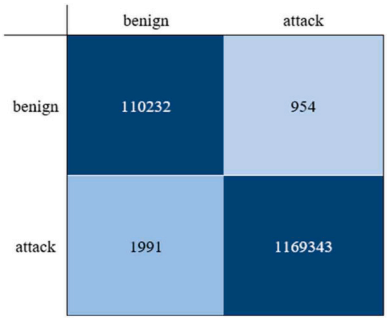

Fig. 3. Confusion matrix for binary classification results using proposed model on n-BAIoT2018 dataset.

Table 3
Comparison of binary classification results over n-BAIoT2018 dataset.

Model	Accuracy	Recall	Precision	F1-Score
AE	92.80	95.58	91.83	93.67
LSTM	94.18	94.83	95.69	95.26
MLP	96.25	96.25	96.26	96.25
Proposed LAE-MLP	99.77	99.92	99.83	99.87

$$b^{(l)} \leftarrow b^{(l)} - \alpha \frac{\partial \mathscr{L}_{\boldsymbol{MLP}}}{\partial \boldsymbol{b}^{(l)}} \tag{19}$$

Inference and Anomaly Detection: For anomaly detection, the trained LSTM and MLP models are applied to the testing dataset(X_{test}). Each instance (x_i) in (X_{test}) undergoes a forward pass through the LSTM and AE, resulting in encoded sequences:

$$\boldsymbol{z}_{\boldsymbol{LAE}} = \text{LAE}\left(h^{(t)}\right) \tag{20}$$

These sequences are then passed through the MLP, which uses its learned weights and biases to predict anomalies based on the encoded features. Once the model is trained and evaluated, it can be deployed for use. There are various methods for deploying a machine learning model, with one common approach being to create a web service that allows users to access the model and make predictions on new data.

3.3. Training parameters

All deep learning models were trained on the same machine with exactly same configuration to create a fair comparison environment. In experimentation, all models were trained over 150 epochs with 16 hidden layers and 32 hidden units (neurons). The experiment was started from the 20 epochs and 4 hidden layers along with 4 neurons then keep increasing them until the models show best results without overfitting. The batch size was kept to 64 for this experiment along with the learning rate of 0.001 and the optimizor used in the experiment was Adam. The parameters were selected based on the empericial experiments for the optimal results.

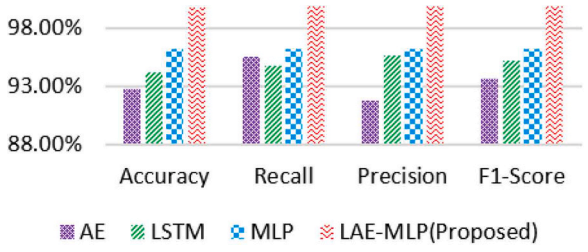

Fig. 4. Comparison of binary classification results over n-BAIoT2018 dataset.

Table 4
Proposed Model results for multi-classification on n-BAIoT2018.

Class	Accuracy	Precision	Recall	F1-score
Normal	99.61	99.70	99.62	99.66
Gafgyt_combo	99.28	99.49	99.76	99.62
Gafgyt_junk	97.65	98.95	98.19	98.57
Gafgyt_scan	99.65	98.84	98.89	98.86
Gafgyt_tcp	99.69	99.46	99.63	99.55
Gafgyt_udp	99.57	99.50	99.18	99.34
Mirai_ack	99.59	99.76	99.15	99.45
Mirai_scan	99.61	99.23	99.02	99.12
Mirai_syn	99.61	99.56	99.22	99.39
Mirai_udp	99.30	99.43	98.96	99.19
Mirai_udpplain	99.56	99.16	98.10	98.62

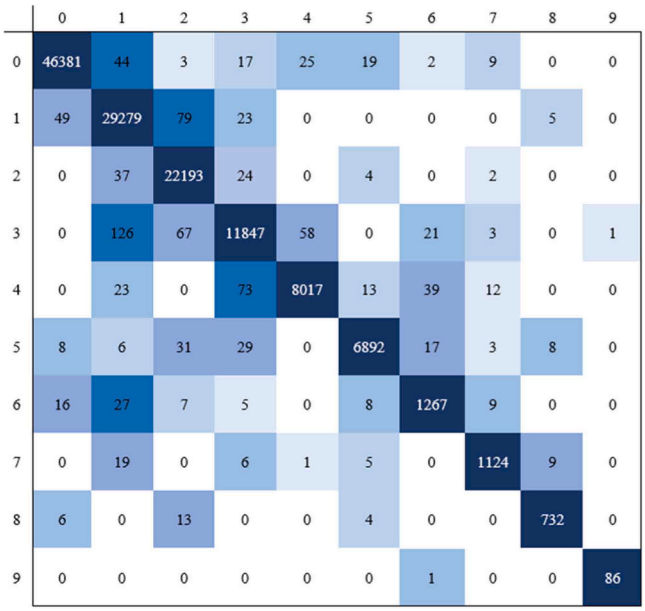

Fig. 5. Confusion matrix for multi-classification results using proposed model on n-BAIoT2018 dataset.

3.4. Evaluation metrics

To measure the results of different models authors used some universal parameters known as performance metrics or evaluation metrics, including confusion matrix, accuracy, precision, recall and f1-score.

4. Results and discussion

The proposed work is applied on two publicly available datasets. This section provides description about those datasets and presents the results of deployed model along with its comparative analysis with other models.

4.1. Results on n-BAIOT 2018 dataset

The N-BAIoT2018 dataset is being used for IoT intrusion detection by many researchers working on IoT security as it encompasses diverse IoT network traffic scenarios. Mentioned dataset consists of normal traffic along with 10 attack classes. It covers attacks such as Mirai, DoS, and brute force attacks. With over 70.6 million instances, it consists of comprehensive data for analysis of different types of attack on many different IoT devices normally used in households or on the commercial level. The distribution of instances among classes varies based on the dataset version, encompassing normal traffic and different types of IoT attacks.

This dataset captures traffic from 9 real-world IoT devices, spanning smart home and network settings to emulate authentic IoT environments. It comprises 115 features, capturing various characteristics of IoT network traffic. The proposed LAE-MLP model significantly outclassed all others, reaching an impressive accuracy of 99.77 % and a near-perfect precision and recall, demonstrating its substantial

Table 5
Proposed Model results for binary classification on UNSW-NB15 dataset.

Metric	Score
Accuracy	99.67
Precision	99.62
Recall	99.78
F1 − score	99.70

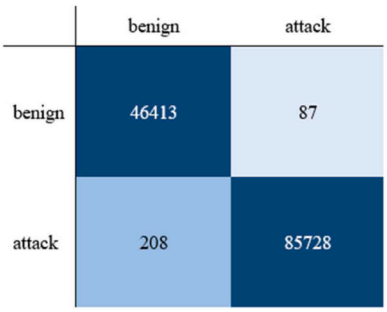

Fig. 6. Confusion matrix for binary classification results using proposed model on UNSW-NB15 dataset.

improvement over the baseline models as illustrated in Table 2 along with the confusion matrix in Fig. 3.

The n-BAIOT 2018 dataset revealed compelling insights into the performance of various anomaly detection models. The autoencoder (AE) demonstrated respectable results with an accuracy of 92.80 %. It displayed a high recall of 95.58 %, indicating its proficiency in identifying actual anomalies. However, its precision of 91.83 % suggested a moderate rate of false positives. In comparison, the LSTM model slightly outperformed the AE, achieving an accuracy of 94.18 % and a precision of 95.69 %, showcasing a superior ability to classify true positives. The MLP model showcased a further improvement with 96.25 % accuracy along with a balanced precision of 96.25 % and recall of 96.26 % as mentioned in Table 3 (Fig. 4).

Along with the binary class, this model was also applied to multi-classification to test the performance and persistence of the model on different attacks classes as it's never been used before, so it was worth applying to both binary and multiclassification. Table 4 shows the multiclass results achieved by proposed model along with the confusion matrix in Fig. 5.

4.2. Results on UNSW-NB 15 dataset

UNSW-NB15 serves as a network intrusion detection dataset and encompasses both normal and attack traffic. It encompasses 9 attack classes along with a normal traffic class and with 2540,044 total instances it provides a rich dataset for studying various attack types. These instances are distributed among different classes, reflecting the prevalence of various attack types in real-world scenarios. While the dataset simulates network traffic, it captures data from multiple devices and sources within a network. The dataset incorporates 49 features, including numerical and categorical attributes, offering comprehensive insights into network traffic, from packet-level information to connection-level data and statistics.

The proposed LAE-MLP model surpassed all other models by achieving an accuracy of 99.67 %. The proposed model's binary classification results for UNSW-NB15 dataset are mentioned in Table 5 along with its confusion matrix in Fig. 6.

While seeing the comparative analysis results for effectiveness in identifying anomalies of UNSW-NB 15 dataset, the Autoencoder (AE) showed an accuracy of 91.38 % and a recall of 93.84 %, showcasing its effectiveness in identifying true anomalies. However, its precision of

Table 6
Comparison of binary classification results over UNSW-NB15 dataset.

Model	Accuracy	Recall	Precision	F1-Score
AE	91.38	93.84	91.59	92.70
LSTM	95.97	96.74	94.13	95.41
MLP	93.51	96.57	92.88	94.69
Proposed LAE-MLP	99.67	99.78	99.62	99.70

Table 7
Proposed Model results for multi-classification on UNSW-NB 15.

Class	Accuracy	Precision	Recall	F1 – score
Benign	99.7	99.8	99.8	99.8
Generic	99.7	99.2	99.4	99.3
Exploits	99.6	99.3	99.8	99.5
Fuzzers	98.8	98.2	97.6	97.9
Dos	98.8	98.1	98.0	98.1
Reconnaissance	99.7	98.2	98.7	98.4
Analysis	98.3	86.6	95.4	90.8
Backdoor	98.3	97.8	96.4	97.1
Shellcode	99.6	98.4	97.9	98.1
Worms	99.9	96.6	99.1	97.8

91.59 % implied a moderate false positive rate. The LSTM model demonstrated notable performance, achieving an accuracy of 95.97 % and a recall of 96.74 %. Additionally, it maintained a balanced precision of 94.13 %, indicating a commendable ability to accurately identify both positive and negative cases. The MLP model also exhibited strong results, attaining an accuracy of 93.51 % and a recall of 96.57 %. However, the proposed LAE-MLP model exhibited remarkable performance over all these models as mentioned in Table 6. Table 7 illustrates the multiclass results for the proposed model over UNSW-NB15 dataset along with the confusion matrix in Figs. 7, 8.

The proposed LAE-MLP model utilizes hybrid deep learning architecture and its experimental results have shown remarkable performance on tested N-BaIoT2018 and UNSW-NB15 datasets through exhibiting accuracy rates of 99.77 % and 99.67 % respectively. These results demonstrate the proposed model potential for enhancing real-world cybersecurity capabilities through accurately detecting botnets in IoT networks. The secret weapon of the LAE-MLP lies in its synergistic fusion of LSTM Autoencoders and Multilayer Perceptrons (MLPs). The LSTM component, adept at discerning temporal patterns within data, extracts salient features with remarkable precision.

This information is then passed to the MLP, a classification expert, which leverages these features to identify botnet attacks with pinpoint accuracy. This dynamic partnership not only surpasses individual deep learning models in performance but also transcends the capabilities of even the advanced state-of-the-art botnet detection techniques. Furthermore, the LAE-MLP's consistency across diverse datasets hints at an exceptional degree of generalizability. This ability to fit to numerous IoT environments and attack scenarios is crucial in continuously evolving threat landscape posed by botnets.

Compared to traditional botnet detection approaches, the LAE-MLP offers several distinct advantages. Firstly, its foundation in deep learning grants it the power to deal with complicated and multi-dimensional nature of IoT network data. Secondly, the incorporation of LSTM Autoencoders empowers it to capture subtle temporal patterns, revealing hidden botnet activity with unparalleled clarity. Lastly, the hybrid architecture, a fusion of LSTM Autoencoders and MLPs, synergistically extracts informative features and performs accurate classification, making the proposed LAE-MLP model a comprehensive and highly effective botnet detection approach.

In short, the proposed LAE-MLP model represents a significant

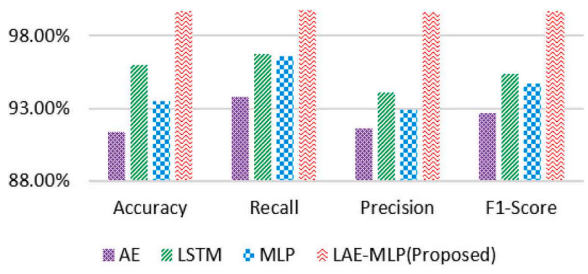

Fig. 7. Comparison of binary classification results over UNSW-NB 15 dataset.

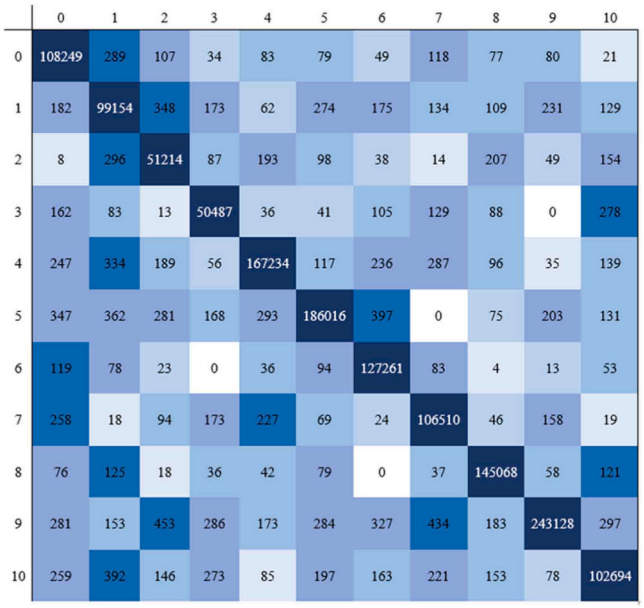

Fig. 8. Confusion matrix for multi-classification results using proposed model on UNSW-NB15 dataset.

paradigm shift in the field of IoT botnet detection. Its exceptional performance, adaptability, and distinct advantages over traditional approaches position it as a beacon of hope for securing our increasingly connected world. The continued research and development efforts exploring new deep learning architectures, incorporating additional data sources, and investigating adversarial attacks will further bolster the LAE-MLP's robustness and effectiveness and hence paving the way for a more resilient future of cyber security in IoT networks.

5. Conclusions and future work

The proposed hybrid model demonstrates substantial potential as a tool for complex data analysis and informed decision-making. Its innovative fusion of latent variable representations with multi-layer perceptrons offers promising capabilities in detecting previously known and zero-day attacks. However, a comprehensive assessment of the model requires acknowledging its limitations, particularly regarding server security, scalability, training complexity, interpretability, and data dependence.

5.1. Limitations

Addressing these limitations will be crucial for ensuring the model's broader applicability and impact. While no system is faultless, the proposed model's reliance on a central server introduces potential security vulnerabilities over time due to the continuous evolution of technological threats. A server compromise could expose sensitive data and model parameters, demanding robust security measures. Furthermore, the model's scalability may demand consideration when being applied on industrial level. Distributed computing and parallel processing techniques may be vital for ensuring its effectiveness in resource-intensive scenarios.

5.2. Future directions

As the future unfolds, the model signals the undertones of improvement. Decentralized AI reinforces security while nurturing collaboration. Sustaining against attacks and server breaches demands advanced encryption. Resource-constrained devices yearn for the model's touch, calling for pruning and lightweight architectures.

Domain-specific wisdom and ensemble techniques hold the key to unlocking performance's hidden chambers. Additionally, the model can be optimized utilizing techniques like pruning or lightweight architectures to deploy on the commercial level for large scale organizations to unlock the full potential of the hybrid LAE-MLP model across a wide range of applications and domains.

Funding

The funding of this work was provided by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2024R719), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

CRediT authorship contribution statement

Shamshair Ali: Writing – original draft, Software, Methodology, Investigation, Conceptualization. **Nauman Qadeer:** Writing – original draft, Visualization, Software, Project administration, Methodology, Conceptualization. **Rubina Ghazal:** Visualization, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Fatimah Alhayan:** Writing – review & editing, Software, Project administration, Methodology, Funding acquisition, Formal analysis, Conceptualization. **Oumaima Saidani:** Writing – original draft, Project administration, Methodology, Funding acquisition, Formal analysis, Data curation. **Rabia Saleem:** Writing – review & editing, Validation, Supervision, Resources, Investigation, Data curation. **Anum Masood:** Writing – review & editing, Project administration, Funding acquisition, Conceptualization. **Deepak Gupta:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Formal analysis. **Muhammad Attique Khan:** Writing – original draft, Supervision, Software, Methodology, Formal analysis, Conceptualization.

Declaration of Competing Interest

All authors declared no conflict of interest in this work.

References

[1] H. Yoon, et al., Trends in internet use among older adults in the United States, 2011–2016, J. Appl. Gerontol. 40 (5) (2021) 466–470.
[2] A. Darem, Anti-phishing awareness delivery methods, Eng., Technol. Appl. Sci. Res. 11 (6) (2021) 7944–7949.
[3] A. Khang, et al., Advanced Iot Technologies and Applications in the Industry 4.0 Digital Economy, CRC Press, 2024.
[4] A. Al-Fuqaha, et al., Internet of things: a survey on enabling technologies, protocols, and applications, IEEE Commun. Surv. Tutor. 17 (4) (2015) 2347–2376.
[5] Cisco. Annual internet report (2018-2023), 2023. (Accessed July 2023). ⟨https://www.cisco.com⟩.
[6] A. Holst, Number of Iot Connected Devices Worldwide 2019-2030 (2022).
[7] G. Vormayr, T. Zseby, J. Fabini, Botnet communication patterns, IEEE Commun. Surv. Tutor. 19 (4) (2017) 2768–2796.
[8] S. Hamzenejadi, M. Ghazvini, S. Hosseini, Mobile botnet detection: a comprehensive survey, Int. J. Inf. Secur. 22 (1) (2023) 137–175.
[9] M. Al-Fawa'reh, et al., MalBoT-DRL: Malware Botnet detection using deep reinforcement learning in IoT networks, IEEE Internet Things J. (2023).
[10] S.S. Silva, et al., Botnets: a survey, Comput. Netw. 57 (2) (2013) 378–403.
[11] J. Zhang, et al., Towards detection of zero-day botnet attack in iot networks using federated learning. in: Proceedings of the ICC 2023-IEEE International Conference on Communications, IEEE, 2023.
[12] Y. Roumani, Patching zero-day vulnerabilities: an empirical analysis, J. Cybersecur. 7 (1) (2021) tyab023.
[13] L. Constantin, Oracle Knew about Currently Exploited Java Vulnerabilities for Months, PC World, 2012.
[14] J. Menn, Microsoft knew about a word bug that put millions of computers at risk but waited 6 months to fix it., in: Business Insider, Business Insider, 2017.
[15] S. Ali, et al., Comparative evaluation of ai-based techniques for zero-day attacks detection, Electronics 11 (23) (2022) 3934.
[16] S. He, et al., Experience report: system log analysis for anomaly detection. in: Proceedings of the IEEE Twenty Seventh International Symposium on Software Reliability Engineering (ISSRE), IEEE, 2016.
[17] M. Al-Qatf, et al., Deep learning approach combining sparse autoencoder with SVM for network intrusion detection, IEEE Access 6 (2018) 52843–52856.
[18] H. Hindy, et al., A taxonomy of network threats and the effect of current datasets on intrusion detection systems, IEEE Access 8 (2020) 104650–104675.
[19] K. Pan, E. Rakhshani, P. Palensky, False data injection attacks on hybrid AC/HVDC interconnected systems with virtual inertia vulnerability, impact and detection, IEEE Access 8 (2020) 141932–141945.
[20] T. Zoppi, et al., On the educated selection of unsupervised algorithms via attacks and anomaly classes, J. Inf. Secur. Appl. 52 (2020) 102474.
[21] Studnia, I., et al. Survey on security threats and protection mechanisms in embedded automotive networks, in: Proceedings of the 2013 Forty Third Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop (DSN-W), IEEE, 2013.
[22] M. Hanselmann, et al., CANet: an unsupervised intrusion detection system for high dimensional CAN bus data, IEEE Access 8 (2020) 58194–58205.
[23] Y. Zeng, X. Hu, K.G. Shin, Detection of botnets using combined host-and network-level information. in: Proceedings of the IEEE/IFIP International Conference on Dependable Systems & Networks (DSN), IEEE, 2010.
[24] Z. Shu, et al., Traffic engineering in software-defined networking: measurement and management, IEEE Access 4 (2016) 3246–3256.
[25] Z. Zhang, et al., Artificial intelligence in cyber security: research advances, challenges, and opportunities, Artif. Intell. Rev. (2022) 1–25.
[26] A. Heidari, M.A. Jabraeil Jamali, Internet of Things intrusion detection systems: a comprehensive review and future directions, Clust. Comput. (2022) 1–28.
[27] A. Alzaqebah, et al., A modified Grey Wolf optimization algorithm for an intrusion detection system, Mathematics 10 (6) (2022) 999.
[28] K. Geetha, S. Brahmananda, Network traffic analysis through deep learning for detection of an army of bots in health IoT network, Int. J. Pervasive Comput. Commun. (2022).
[29] M. Latah, L. Toker, An efficient flow-based multi-level hybrid intrusion detection system for software-defined networks, CCF Trans. Netw. 3 (3) (2020) 261–271.
[30] V. Kumar, D. Sinha, A robust intelligent zero-day cyber-attack detection technique, Complex Intell. Syst. 7 (5) (2021) 2211–2234.
[31] M.S. Alzahrani, F.W. Alsaade, Computational intelligence approaches in developing cyberattack detection system, Comput. Intell. Neurosci. 2022 (2022).
[32] R. Malik, et al., An improved deep belief network IDS on IoT-based network for traffic systems, J. Adv. Transp. 2022 (2022).
[33] M.A. Khatun, N. Chowdhury, M.N. Uddin, Malicious nodes detection based on artificial neural network in IoT environments. in: Proceedings of the Twenty Second International Conference on Computer and Information Technology (ICCIT), IEEE, 2019.
[34] N. Koroniotis, N. Moustafa, J. Slay, A new Intelligent Satellite Deep Learning Network Forensic framework for smart satellite networks, Comput. Electr. Eng. 99 (2022) 107745.
[35] X. Qu, et al., Mfgan: multimodal fusion for industrial anomaly detection using attention-based autoencoder and generative adversarial network, Sensors 24 (2) (2024) 637.
[36] A. Al-Obaidi, A Ibrahim, A.M. Khaleel, The Effectiveness of Deploying Machine Learning Techniques in Information Security to Detect Nine Attacks: UNSW-NB15 Dataset as A Case Study (2023).
[37] M.N. Al-Mhiqani, et al., A new intelligent multilayer framework for insider threat detection, Comput. Electr. Eng. 97 (2022) 107597.
[38] M. Tan, et al., A neural attention model for real-time network intrusion detection. in: Proceedings of the IEEE Forty Fourth Conference on Local Computer Networks (LCN), IEEE, 2019.
[39] M. Abdallah, et al., A hybrid CNN-LSTM based approach for anomaly detection systems in SDNs, in: Proceedings of the Sixteenth International Conference on Availability, Reliability and Security (2021).
[40] Said Elsayed, M., et al. Network anomaly detection using LSTM based autoencoder, in: Proceedings of the Sixteenth ACM Symposium on QoS and Security for Wireless and Mobile Networks. 2020.