


Simultaneous Botnet Attack Detection Using Long Short Term Memory-Based Autoencoder and XGBoost Classifier



Soundes Belkacem 

LAMIE Laboratory, Department of Computer Science, University of Batna 2, Batna 05078, Algeria

Corresponding Author Email: s.belkacem@univ-batna2.dz

Copyright: ©2024 The author. This article is published by IIETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/ijssse.140115>

ABSTRACT

Received: 29 December 2023

Revised: 8 February 2024

Accepted: 12 February 2024

Available online: 29 February 2024

Keywords:

botnet detection, internet of things (IoT), extreme gradient boosting (XGBoost), long short-term memory (LSTM), autoencoders (AE), NCC-2 dataset

Botnet is a cyber-attack that aims to compromise the security of internet of things (IoT) networks by exploiting infected devices to launch attacks and gain unauthorized access to private information. Intrusion detection system (IDS) emerges as critical countermeasures for tackling the risks posed by botnet attacks, playing a crucial role in ensuring the integrity and confidentiality of data in IoT environments. Developing an effective botnet detection system depends on efficient contextual understanding and accurate attack pattern characterization. Recently, deep learning and machine learning based IDS have demonstrated promising results in traffic pattern recognition and identification as normal or malicious from raw data. However, these approaches fail to detect simultaneous botnet attacks as it ignores its distributed nature. In this paper, we propose an efficient hybrid deep learning model for simultaneous botnet attack detection over IoT networks. The two-stage hybrid model analyzes the network traffic data captured from three parallel sensors and extracts simultaneous characteristics of attack traffic. The use of parallel detection enables more comprehensive coverage of the network, thereby increasing the detection accuracy of malicious activities that could be missed by a single sensor. Features are extracted using a long-short-term memory base autoencoder (LSTM-AE) over the NCC-2 Simultaneous Botnet Dataset. The LSTM-AE is trained using data from multiple sensors to model temporal characteristics and results in reduced latent representation. Attack type identification is achieved through a multi-class classification using the Extreme Gradient Boosting (XGBoost) ensemble learning algorithm. The recently released NCC-2 dataset is the first dataset to provide data representing sequential and simultaneous botnet activities detected concurrently by multiple sensors. Performance exploration indicates that for parallel botnet detection, the proposed LSTM-AE-XGB model achieves high accuracy while reducing false or missing detection. Moreover, to demonstrate model efficiency, we conducted a 10-fold cross-validation and a comparative performance analysis with the state-of-the-art ML and DL-based techniques for feature extraction and simultaneous botnet detection.

1. INTRODUCTION

The emergence of Internet of Things (IoT)-related services has become a revolutionizing technological development that significantly impacts various aspects of our daily lives. With over ten billion interconnected devices via the Internet, IoT-based application usage is continuously growing in many fields, including healthcare, education, agriculture, transport, and industry [1, 2]. IoT device communication often requires the generation and exchange of a substantial volume of private data. However, hardware limitations and insufficient security arrangements of IoT devices contribute to their vulnerabilities against cyber-attacks and fail to ensure a secure data transaction [3]. Cyber-attacks are illegitimate actions carried out by an external agent that involve unauthorized attempts to access confidential data or perform malicious actions through an infected device.

IoT botnet cyber-attack are considered a prominent threat to

IoT devices [4]. In 2021, the FBI's Internet Crime Complaint Center (IC3) received more than 6.9 billion reports of cybercrime [3]. A botnet is a compromised network utilized by a botnet master to launch attacks or illegal actions [5], causing data loss and even threatening the usability of the entire network. Therefore, the early detection of botnet activity using an intrusion detection system (IDS) is crucial for network security. Existing solutions for botnet detection may be categorized into signature-based and anomaly-based methods. Signature-based methods apply pattern matching of traffic characteristics to a predefined signature [6]. However, those methods required the regular updating of the signature database by a cyber security expert. In addition, it fails to detect zero-day attacks. Anomaly-based intrusion detection systems (IDS) are a recent and efficient countermeasure to tackle botnet attacks [7]. Anomaly-based IDS analyzes network traffic data to classify it as normal or malicious (Figure 1). IDSs aim to detect compromised nodes before any

malicious actions are initiated and adopt real-time countermeasures to ensure data integrity and protection [8].

Recently, machine learning (ML) and deep learning (DL) techniques have demonstrated their efficiency and robustness for cyber-attack detection and prevention over IoT networks. Techniques such as convolutional neural networks (CNN) [9], deep neural networks (DNN) [10], long short-term memory (LSTM) [11], extreme gradient boosting (XGBoost) [12], autoencoders [2], multi-layer perceptron (MLP) [13], and random forest (RF) [14]. Anomaly botnet detection models based on machine learning and deep learning are trained to recognize complex patterns in network traffic through the use of learned characteristics. An indication of malicious activity may be detected if a deviation from normal network behavior is detected.

Existing solutions have demonstrated major drawbacks. They have mostly focused on a binary classification problem for discriminating between normal and illegitimate traffic. However, they fail to identify specific attack types, which is a crucial step for network security enhancement and threat prevention. In fact, attack type identification enables the development of effective detection methods and defense mechanisms adjusted to specific attack. Furthermore, the absence of a realistic dataset that considers the nature of botnet attacks as a distributed activity. While it is classified as a distributed or spread activity, existing datasets and solutions don't consider simultaneous data or parallel detection. Recently, the NCC-2 dataset [15] was released to address the lack of these fundamental characteristics. To overcome these limitations, we propose a LSTM-AE-XGB model that makes use of parallel sensor data to identify the patterns of botnet attack activities. Our approach improves detection accuracy by applying a long-short-term memory base autoencoder (LSTM-AE) algorithm for feature extraction to extract patterns, correlations, and dependencies within network traffic. Attack type identification is performed through a multi-class classification using the Extreme Gradient Boosting (XGBoost) classifier.

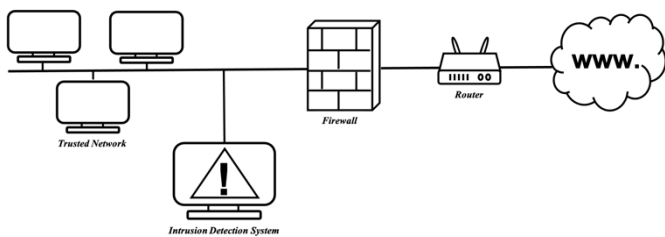


Figure 1. A network intrusion detection system

We use a long-short-term memory-based autoencoder (AE-LSTM) to extract and generate a new feature set representing attack classes independently of instance numbers; this is essential when dealing with imbalanced data. We train AE-LSTM over data from multiple sensors to extract temporal dependencies and long-term patterns in attack and normal traffic data. Computed latent representation increases the distance over samples derived from dissimilar classes to enhance the classification performance. Based on the computed latent feature representation, we perform a multiclass classification task using the XGBoost Classifier. In this study, we explore LSTM-AE-XGB model performances using the recently released NCC-2 Simultaneous Botnet dataset. In comparison to existing datasets that do not include such valuable information. The dataset includes data from

multiple parallel sensors, consisting of both normal traffic and sequential and simultaneous botnet attack activities.

The main contributions of this paper are:

- We propose an efficient and accurate simultaneous botnet attack detection and identification model over combined traffic from multiple sensors.
- LSTM-AE feature selection and extraction approach was used to automate the selection and extraction of latent representations of temporal dependencies, long-term patterns of attack and normal traffic.
- We performed an extensive performance evaluation of the proposed model using the recently released NCC2dataset.
- Experiments reveal that the proposed hybrid model is able to accurately detect and identify various traffic activities.
- We conducted a comparative analysis of feature extraction and parallel botnet detection approaches based on ML and deep learning.

The rest of this paper is organized as follows. In section 2, the recent solutions for IoT botnet attack detection are investigated. Section 3, gives a detailed description of the NCC-2 Dataset. In section 4, proposed methodology for the IoT-Botnet detection and identification is introduced. The experimental results are presented and discussed in section 5. The paper is concluded with future research scope in section 6.

2. PREVIOUS WORKS

In this section, we explore recent research papers focusing on ML and DL-based IoT IDS. We applied a methodological categorization that groups studied solutions into five categories based on the approaches used: convolutional neural networks (CNNs), long short-term memory (LSTM), XGBoost classifier, autoencoders (AE), and random forest (RF) based models.

The CNN and negative selection algorithms are used to first identify traffic packets as bots or not and then identify the botnet type [16]. The model achieves an accuracy of 99% over the ISCX 2012 dataset. An integrated deep neural network and principal component analysis (PCA)-based method for zero-day attack detection is proposed by Al-Fawa'reh et al. [10]. Model analyze flow data in cloud environments for real-time anomalies' detection. It's reported that the PCA-DNN model achieves an accuracy of 98% for all attack detection over the cloud-based dataset. Hoang and Vu [14] aim to detect DGA botnets by enhancing the existing ML detection model. Increasing the detection rate and decreasing false alarms are achieved through the addition of seven new domain features. The model gives results of 3.02% false alarm rate and an overall accuracy of 97.03%.

An XGBoost based network intrusion detection system is proposed by Mohiuddin et al. [17]. A reduced feature map is generated using a modified wrapper-based whale sine-cosine method (MWWSCA). A binary and multiclass classification attack detection is performed using a weighted extreme gradient boosting (XGBoost) classifier. Experimental results of binary classification over two datasets, UNSW-NB15 and CICIDS-2017 revealed an accuracy of 99% and 98% respectively, while 91% for multiclass classification over the UNSW-NB15 dataset. The XGBoost classifier has been used for compromised IOT device detection [18]. The model achieved an accuracy of 93.6% of the IoT-23 dataset. Li et al. [19] investigate the uses of several generative adversarial network (GAN) architectures for the generation of realistic

network traffic instances. Synthetic traffic samples are classified using the XGBoost classifier. The Wasserstein GAN architecture achieves the best overall performances with 99.97%, 99.94%, and 100% for accuracy, recall, and precision, respectively, over the UNSW NB15 dataset.

The LSTM network has been widely used for intrusion detection from long traffic data. Its ability to learn and discriminate data flow instances based on sequential network traffic behavioral characterization is crucial to IDS performance. CNN and LSTM networks are combined for real-time detection of DDoS based on local and temporal features of traffic obtained from four security cameras [11]. A hybrid model, DNN-LSTM that could target cyber-attacks is proposed by Sattari et al. [5]. The two DL networks are executed simultaneously over the training dataset, and the later obtained results are merged for traffic type identification. The method is able to detect 99.98% of bot attacks within 0.022 ms over the N BaIoT dataset. An edge-assisted anomaly-based architecture for IoT attack detection is proposed [19]. Multi-edge collaborative traffic capture and y detectors dedicated to each type of device dedicated LSTM autoencoder detector for each specific device.

IDS aims to characterize attack activities through the analysis of large-scale traffic datasets. To enhance classification performances, recent solutions use ML and DL techniques for feature selection and extraction. Autoencoders (AE) have been widely used to reduce data complexity and generate an optimal descriptive representation. Haseeb et al. [2] propose an AE-based feature extraction method to cluster IoT attacks. Attack behavior is identified from changes in command data and generates a feature representation based on the correlation between commands. The K-means-clusters $k = 8$ classifier is used to regroup IoT attacks within clusters. Evaluation shows that attacks with common features are grouped within the same cluster. AE and triplet networks are combined by Andresini et al. [20] for feature extraction from imbalanced data. The triplet network is used over the generated AE representation to associate each flow with its closest reconstruction class. The use of latent representation enhances classification performance by considering AE embedding learning. The evaluation results over three datasets, KDDCUP99, AAGM17, and CICIDS17 is 93.50%, 89.63% and 98.24% respectively. DoS and DDoS attacks were detected based on variational AE [21]. Latent representation is learned from malicious and benign traffic. The authors proposed two VAE-based methods. The first is used as a binary classifier for two traffic types. The second one uses VAE trained on legitimate traffic to filter out the anomalies. Two methods were tested over the CSECICIDS2018 and CICIDS2017 datasets. The obtained results show the efficiency of both methods, with slightly higher performances for the classifier-based method in comparison to the anomaly-based method. In the study of Bârli et al. [22] an optimized lightweight AE IoT anomaly detection model is introduced. The model consists of a two-layer microswarm population-based optimizer that simultaneously selects features and autoencoder neurons. The KNN classifier is trained using the selected feature representation with the computed AE complexity. Experiments are conducted over the N-Baiot dataset and show the highest performances of the proposed model in comparison to existing non-optimized and well-known optimizer-based solutions. The reported accuracy is 99%, with a complexity of 2 nerines and a feature dimension of 30. A solution for new and unknown cyber-attack detection

in the IoT network is introduced by Vu et al. [23]. Three regularized variants of AEs are proposed to accurately predict unknown attack features. AEs variants, namely MDA, MVAE, and MAE are trained on two traffic types: Normal and unknown attacks, and generate a latent feature representation that will be used for the training of four classifiers: SVM, LR, PCT and NCT. Methods performances are investigated on nine datasets and demonstrate that proposed models enable the detection of unknown attacks through the projection of initial features into a linear and isolated latent representation. The three proposed methods outperform existing models that use either the original or AE and DBN-generated feature sets over all the datasets used.

Existing IDS discriminates between two types of traffic: benign or malicious. Recently, researchers have focused more on attack type identification to ensure the adoption of the appropriate prevention technique. Thakur et al. [24] propose a generic-specific AE architecture to first learn the common characteristics of intrusion activities and then learn those related to each specific attack type. The CICIDS 2017 dataset is used for model evaluation. The method achieves a high accuracy of up to 1 for the detection of 14 attack types, including DDoS, Slowloris, Slowhttptest, Hulk, GoldenEye, Heartbleed, PortScan, Bot, FTP, SSH, Brute Force, XSS, SQL Injection, and Infiltration.

The Random Forest (RF) classifier is combined information gain (IG) for relevant feature extraction in the study of Yin et al. [13]. The resulting feature set is processed through a machine wrapper method that provides a recursively eliminated feature based on learning-based techniques. The MLP classifier is used for multi-classification of attack types. The obtained experimental results over the UNSW-NB15 dataset are 84.24% over the 23- feature subset. Khan and Mailewa [25] identify four attack classes over the NSL-KDD dataset. Deep AE and support vector machine (SVM) are combined for attack feature extraction and detection from non-linear high-dimensional feature space. Method evaluation indicates that the DAE+SVM model has high performance metrics in comparison to PCA+SVM. The authors find that low-frequency attacks are detected with a 0.72 micro-average score, and the L1 regularization technique is more efficient than Lasso regularization over the used dataset. Features set minimization for ML-based intrusion detection is investigated in the study of Kalakoti et al. [26] for binary and multiclass classification for the detection of IoT botnet attacks using two datasets, Med-BIoT and N-BaIoT. Filter and wrapper methods SFS and SBS are used to generate optimal feature representation to guarantee high detection accuracy for each classification problem. It was shown through experiments that post-attack detection is related to channel-based features, while bot attacks are influenced by host-based features. The authors indicated that they have high detection performance using various machine learning classifiers.

3. PROPOSED METHODOLOGY

3.1 Problem formulation

Let dataset D , consist of three sub-datasets:

$$D = \{D_1, D_2, D_3\} \quad (1)$$

Each includes data captured from Sensor S1, Sensor S2 or

Sensor S3 respectively. Each sub-dataset D_i contains $\sum_{k=1}^{M_i} S_k$ samples, each sample consist of a total of 18 features:

$$S_k = \{f_{k1}, f_{k2}, \dots, f_{k15}, ActivityLabel_k, BotnetName_k, ensorId_k\} \quad (2)$$

where, *ActivityLabel* represents traffic type as botnet or normal. The feature *BonetName* represents the botnet activity type as *Neris*, *Virut*, *Rbot*, *Nsys.ay*, *Sogo*, *Murlo*, *Menti*, or *Normal*. The *ensorId* feature represents the sensor identifier as 1, 2, or 3. In this paper, we aim to detect the attack type of each traffic sample using the LSTM-AE-XGB model. We intend to use hybrid DL model to address a multiclass classification problem for botnet activity identification over the NCC2 dataset. A long-short-term memory-based autoencoder (LSTM-AE) is utilized to extract and generate a new feature set representing attack classes. The computed latent feature representation captures temporal dependencies and long-term patterns present in traffic data captured simultaneously over parallel sensors. Traffic activities are detected and classified using the XGBoost classifier.

A general overview of the proposed model LSTM-AE-XGB architecture for simultaneous botnet attack detection is given in Figure 2.

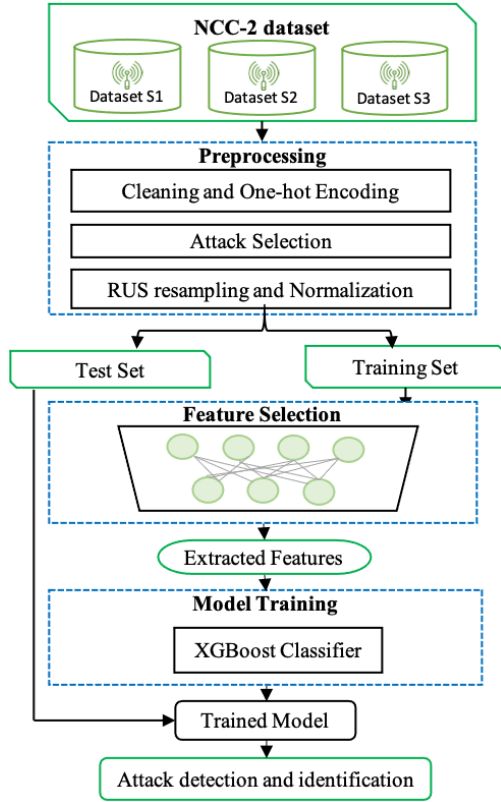


Figure 2. An overview of the LSTM-AE-XGB model for simultaneous botnet attack detection

3.2 The NCC-2 dataset simultaneous botnet dataset

In this paper, the recently released NCC-2 Simultaneous Botnet dataset [15] is used for feature extraction and model training. The dataset includes data from three parallel sensors that capture normal, sequential and simultaneous botnet attack activities. It provides a traffic network taking into consideration the distributed nature of botnet attacks. In this dataset, botnet attack scenarios are simulated based on attacks

extracted from the CTU-13 [27] and NCC [28] datasets. The two types of network traffic activities Botnet and normal are simultaneously captured over three sensors. The dataset contains a total of four sub-datasets. Three subsets include traffic data from each sensor S1, S2 and S3 recorded over eight hours. In addition, a sub-dataset incorporates a combination of those subsets.

The sub-dataset contains traffic captured by sensor S1 consists of several botnet activities, including *Neris*, *Rbot*, *Virut*, *Sogo* and *NSIS.ay*. Sensor S2 sub-datset includes traffic from four attack activities: *Menti*, *Virut*, *Rbot* and *Neris*. Traffic data collected from the third sensor S3 includes botnet types: *Virut*, *Rbot*, *NSIS.ay*, *Neris*, and *Murlo* botnet attack types. A combination of those three sub-datasets is also provided. Table 1 lists the instance numbers for each botnet attack type captured from different sensors in the NCC2 dataset.

The dataset contains 15 features existing in the CTU-13 and NCC datasets, including StartTime, SrcAddr, Activity time, Dur, DstAddr, Proto, Sport, Dport, Dir, State, dTos, sTos, TotPkts, SrcBytes, TotBytes and Label. In addition, there are three features describing traffic type, botnet activity type, and sensor information.

Table 1. Traffic type distribution for the three sub-datasets of the NCC-2 dataset

Sensor	Traffic type	Instances	Total
Sensor S1	Neris	47000	146000
	Rbot	62000	
	virut	19000	
	Nsis.ay	9000	
	Sogo	9000	
	Normal	4749158	4749158
Sensor S2	Neris	267000	364000
	Rbot	72000	
	Virut	19000	
	Menti	6000	
	Normal	5634133	5634133
Sensor S3	Neris	220000	294000
	Rbot	13000	
	virut	38000	
	Nsis.ay	9000	
	Murlo	14000	
	Normal	3591792	3591792

3.3 Preprocessing

The NCC2 dataset is initially preprocessed by applying data cleaning, attack selection, and encoding techniques. The used data set consists of three sub-datasets. Each corresponds to network traffic captured by a sensor. Captured data depends on the covered segment of the network. This includes specific criteria such as a defined range of IP addresses, ports, and particular protocols used. Hence, the different sub-datasets are preprocessed separately. Preprocessing involves cleaning, encoding, attack selection, resampling, and normalization. Data is cleaned by removing null values, duplicate rows, and unused features such as the binary classification label *ActivityLabel*.

The proposed model aims to detect attacks simultaneously captured by all three sensors; each attack class that is not captured in parallel is removed. As a result, the feature *BonetName* will represent five attack types: *Neris*, *Virut*, *Rbot*, *Nsis.ay*, and *Normal*, and the model is used for five class classification task.

Despite the fact that the performances of ML and DL models are strongly affected by imbalanced data, Moreover, the Normal class represents the majority class with 13975083 instances, while the total number of botnet attack instances is significantly lower. We apply majority class undersampling using the Random Undersampling Technique (RUS) to balance normal and attack traffic activities at 50% each.

Finally, the three sub datasets are merged into one dataset, and the Z-score normalization technique is applied to scale the data. The resulting dataset is then split with a ratio of 80% for training and 20% for tests used for classifier training and classification.

3.4 Feature extraction

Relevant features from the dataset are extracted and reduced using LSTM-AE. In fact, efficient feature extraction is crucial for enhancing classification results in high-dimensional datasets.

Autoencoder (AE) is an unsupervised learning artificial neural network. Intended to generate a low-dimensional feature representation of the input data in a nonlinear space [29]. The architecture of an autoencoder consists of three layers: the input layer, the encoder, and the decoder neural network. The encoder processes the input data to produce a learned latent representation that captures key features. In contrast, the decoder aims to precisely reconstruct the input data based on the produced representation.

The LSTM-based autoencoder (LSTM-AE) is widely used for anomaly detection, dimensionality reduction, and significant characteristic extraction from series data [29-31]. In this work, we use the LSTM neural network as the encoder layer. LSTM-AE aims to model temporal dependencies over the traffic sequence and produce an optimal latent representation. We train a one-layer LSTM-based autoencoder using the preprocessed data to generate a latent representation L computed using the following equation:

$$L = LSTM(WS + b) \quad (3)$$

where, S is the traffic data sequence from the input layer, W is the weight matrix, and b is the bias vector.

The LSTM-AE consists of one LSTM layer of 32 units and a ReLU activation function, followed by a RepeatVector layer. The Adam optimizer and the Mean Squared Error (MSE) loss function are used for model compilation. The training is performed across 10 epochs with a batch size of 256. Algorithm 1. gives a description of the LSTM-AE-based feature extraction method. The produced latent representation L is considered an optimal feature representation and will be further considered for classifier training and evaluation.

3.5 Attack detection and identification

Extreme Gradient Boosting (XGBoost) is an ensemble learning technique that employs the gradient boosting technique over a set of decision trees. The XGBoost classifier presents strong advantages, including the ability to effectively classify instances by learning from previous mistakes, employ fine-tuning hyperparameters, adopt regularizing techniques, and handle imbalanced data [32].

Algorithm 1. Pseudo code of LSTM-AE based feature extraction method

Input:

- X_{train} : Preprocessed training subset
- X_{test} : Preprocessed test subset

Output:

- L_{train} : Latent representation for the training subset
- L_{test} : Latent representation for the test subset

Local parameters:

- $X_{train} \leftarrow (1, 14)$
- Latent_dim $\leftarrow 8$
- Number_epochs $\leftarrow 10$
- batch_size $\leftarrow 256$
- Activation_function $\leftarrow Relu$
- Used_optimizer $\leftarrow Adam$
- Selected_loss_function $\leftarrow MSE$

Begin

```

create encoder_model
encod_model.add(LSTM layer)
encoder_model.add(RepeatVector layer)
encoder_model.compile(optimizer='adam', loss='mse')
encoder_model.fit(X_train)
L_train  $\leftarrow$  encoder_model.predict(X_train)
L_test  $\leftarrow$  encoder_model.predict(X_test)

```

End

3.6 Attack detection and identification

Extreme Gradient Boosting (XGBoost) is an ensemble learning technique that employs the gradient boosting technique over a set of decision trees. The XGBoost classifier presents strong advantages, including the ability to effectively classify instances by learning from previous mistakes, employ fine-tuning hyperparameters, adopt regularizing techniques, and handle imbalanced data [32].

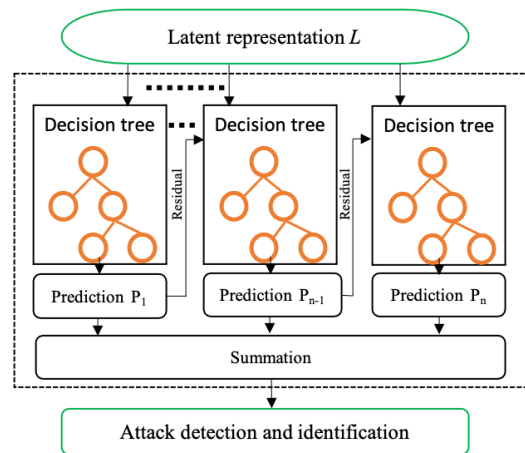


Figure 3. The overall architecture of the used XGBoost classifier

XGBoost aims to reduce prediction errors through the residual aggregation of weak learner (decision tree) at each step. To ensure low complexity and avoid overfitting, a regularization technique is employed [18]. It aims to minimize the loss following the equation:

$$L(t) = \sum_{k=1}^n l(y_k, y_k^{t-1} + f_t(X_i)) + \theta(f_t) \quad (4)$$

where, y_k^{t-1} is the prediction of the k example by the increment t and $\theta(f_t)$ is a penalty parameter [18].

In this paper, we train the XGBoost classifier using the computed latent representation L . The architecture of the XGboost classifier is given in Figure 3. We aim to accurately detect anomalous activity and identify the exact botnet type.

3.7 Evaluation metrics

Extreme Gradient Boosting (XGBoost) is an ensemble learning technique that employs. In this work, we address a multiclass classification problem using ML and DL techniques. Hence, for the proposed model evaluation, the standard metrics, including accuracy, precision, recall, F1-score and the confusion matrix (CM) are considered. To compute these measures, four parameters are first extracted: true negative (TN), true positive (TP), false negative (FN) and false positive (FP). and false negative (FN). The TP, TN represent traffic type instances correctly identified, and the FP, FN are the miss-classified instances. The following equations are used to compute the following metrics:

$$Accuracy = \frac{(TP+TN)}{(TN+TP+FN+FP)} \quad (5)$$

$$Precision = \frac{TP}{FP+TP} \quad (6)$$

$$Recall = \frac{TP}{FN+TP} \quad (7)$$

4. EXPIREMETAL RESULTS

The performance of the proposed AE-LSTM-XGB model for parallel botnet detection is explored and presented in this section. Moreover, we performed a comparative performance analysis of the state-of-the-art ML-based and DL-based solutions for efficient feature extraction and parallel botnet detection.

4.1 Experimental setup

Table 2. Parameters of the used XGBoost classifier

Parameters	Description	Value
Evaluation Metric	Metric for measuring performance during training	Log loss function
Learning Rate	Model weight update step size	0.3
Objective	Probabilistic classification of multiclass data	-
Gamma	Leaf splitting minimum reduction loss	0
Random State by Tree	Reproducibility random seed	0
Regularization Lambda	Term of L2 regularization	1
Max Depth	Tree maximum depth	6
Iterations (n_estimators)	Training rounds	100
Scale POS Weight	Scaling positive examples errors	1
Min Child Weight	Leaf node minimum required sum of weights	1
colsample_bytree	Features percentage for tree	1
Subsample	Training instances percentage used by each tree	1

The experimental setup used the Google Colaboratory environment. Google Colab is a cloud-based Jupyter notebook

environment that incorporates required software, including Python 3, and uses libraries such as Scikit-learn, Keras, NumPy, Pandas, and TensorFlow. The XGBoost classifier is trained with the parameters listed in Table 2.

4.2 Performance analysis

To ensure accurate detection over the NCC2 dataset, we processed data to balance traffic class instances. We have applied the RUS downsampling methods to balance sample data for different traffic activities. Figure 4 shows the class distribution over all three sensors before and after class balancing. The network activity type is 50% for normal and 50% for total attack traffic types.

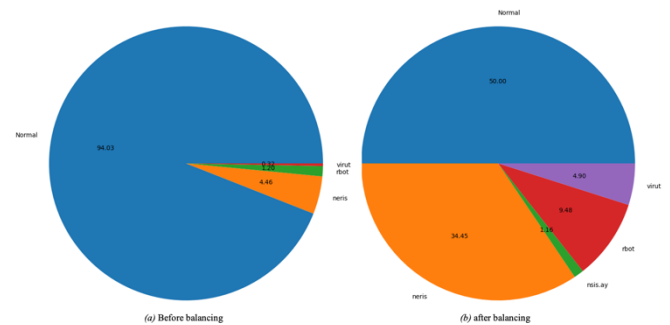


Figure 4. Class distribution of the NCC2 dataset before and after preprocessing

The performance of the proposed model is presented in Table 3 and the confusion matrix is given in Figure 7. Results indicate that the proposed model achieves high performance on the used dataset with an F1-score, recall, and precision of up to 0.999.

Table 3. Multi-class classification report of the proposed model over the NCC2 Dataset

Class Name	Pre	Rec	F1	support	Class freq %
Normal	1.000	1.000	1.000	129149	50.00
Neris	1.000	1.000	1.000	92940	35.89
Rbot	0.999	0.998	0.999	13096	05.11
Virut	0.998	0.999	0.999	13156	05.10
Nsis.ay	1.000	1.000	1.000	10126	03.89
Accuracy		0.999		258467	0.0
Macro avg	0.999	0.999	0.999	258467	0.0
Weighted avg	1.000	1.000	1.000	258467	0.0

The XGB classifier achieves an accuracy, precision, and recall of 1,000 on the “Normal, Neris” and “Rbot” classes, indicating the model’s efficiency in correctly detecting class instances from these classes with minimal misclassification. These results demonstrate that the proposed model is highly effective and can accurately distinguish instances, even for small classes. The macro average value of 0.999 and weighted average value of 1,000 indicate the proposed model’s ability to efficiently learn important features of different attack types and emphasize its reliability for imbalanced multiclass classification.

Table 4 lists the results of 10-fold cross-validation of the proposed LSTM-AE-XGBoost model. The model demonstrates an average accuracy of 0.9997, and F1 scores up to

0.9998 across 10 folds. Precision values range from 0.999623 to 0.999845. The model presents significant efficiency across all folds, indicating its ability to accurately predict attack types with minimal misclassification. Proposed model consistent performances suggest its robustness and generalization capacity.

Table 4. 10-fold cross validation of the proposed model LSTM-AE-XGBoost

Fold	Acc	Pre	Rec	F1
1	0.999845	0.999845	0.999845	0.999845
2	0.999778	0.999845	0.999778	0.999778
3	0.999836	0.999836	0.999836	0.999836
4	0.999758	0.999759	0.999758	0.999758
5	0.999642	0.999643	0.999642	0.999642
6	0.999836	0.999836	0.999836	0.999836
7	0.999816	0.999816	0.999816	0.999816
8	0.999623	0.999623	0.999623	0.999623
9	0.999826	0.999826	0.999826	0.999826
10	0.999826	0.999826	0.999826	0.999826
μ	0.999779	0.999786	0.999779	0.999779
σ	0.000082	0.000084	0.000082	0.000082

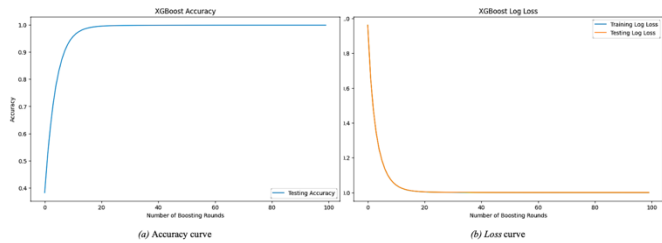


Figure 5. Accuracy and loss curves of the proposed model over the NCC2 dataset

Accuracy and loss curves are presented in Figure 5. The proposed model presents a stable curve over epochs. Accuracy increases to 0.999 around epochs ranging from 0 to 20. Loss curves decrease in the first 20 epochs and remain stable near 0.1. The observed patterns demonstrate that the proposed model learns effectively and converges efficiently throughout the training process, resulting in high accuracy with minimal loss.

4.3 Evaluation of feature extraction technique

Preprocessed data is subjected to feature extraction and reduction. In the proposed model, an LSTM-based AE is used to compute an optimal latent representation. To demonstrate the importance of relevant feature extraction methods for attack detection and their ability to affect classification performances, we compared the performances of feature extraction method to efficient used in IDS.

For comparison, the LSTM-AE is compared to LSTM and random forest (RF)-based techniques. The XGBoost classifier is trained using different feature sets, while the preprocessing step is unchanged for the three techniques. The RF-based method selects features based on their importance. The optimal feature set is generated by retaining relevant features based on a high score [13].

Figure 6 shows an evaluation of classification metrics and indicates that the LSTM-AE technique achieves the best results in comparison to other feature extraction techniques. The XGBoost classifier exhibits a high precision of 0.999 over

the LSTM-AE set, implying a low missed identification. The lowest results are given using LSTM-based extracted features set with a precision of 0.913.

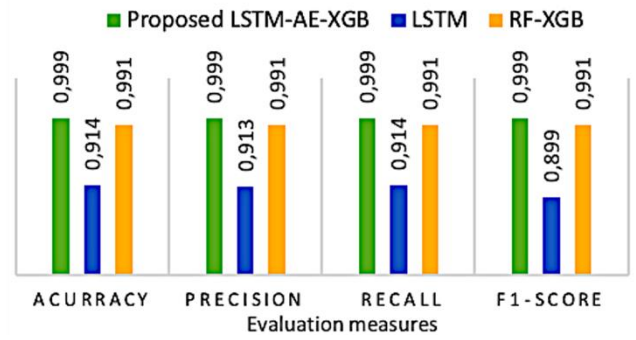


Figure 6. Evaluation of different feature extraction techniques

4.4 Evaluation of attack detection and identification

IDS intends to increase the number of detected malicious activities and reduce the number of reported attacks. Hence, precision and F1-score metrics are considered accurate indicators of model performance and efficiency. For further evaluation, we compare the results obtained by the proposed model with three classifiers.

Using the computed latent representation, we trained random forest (RF), CNN-LSTM, and multi-layer perceptron (MLP) for simultaneous parallel botnet attack detection and identification. Table 5 lists metrics comparisons of different classifiers over the NCC2 dataset. Figure 7 presents the confusion matrix of four classifiers over the test set.

Table 5. Metrics comparison of proposed model with other classifiers using LSTM-AE features over the NCC2 dataset

Classifier	Class Name	Pre	Rec	F1
CNN-LSTM	Normal	0.999	0.997	0.998
	Neris	0.997	0.998	0.997
	Rbot	0.962	0.983	0.973
	Virut	0.968	0.948	0.958
	Nsis.ay	0.974	0.978	0.976
RF	Normal	0.932	0.999	0.965
	Neris	0.946	0.990	0.968
	Rbot	0.721	0.435	0.542
	Virut	0.768	0.401	0.527
	Nsis.ay	0.977	0.775	0.864
MLP	Normal	1.000	0.999	0.999
	Neris	0.995	0.999	0.997
	Rbot	0.976	0.917	0.945
	Virut	0.947	0.986	0.966
	Nsis.ay	0.989	0.990	0.989
Proposed LSTM-AE-XGB	Normal	1.000	1.000	1.000
	Neris	1.000	1.000	1.000
	Rbot	0.999	0.998	0.999
	Virut	0.998	0.999	0.999
	Nsis.ay	1.000	1.000	1.000

Most classifiers demonstrate high performance for the detection of “Normal” and “Neris” traffic activities, with a precision ranging from 0.932 to 1.0, a recall score of 0.917 to 0.999, and a F1-score of about 0.970, indicating their ability to successfully detect both benign patterns and common attack behaviors. However, the CNN-LSTM, RF, and MLP show difficulties in identifying “Rbot” and “Virut” activities. The lowest F1-score is achieved by RF on “Nsis.ay” class indicating detection difficulties for less frequent attacks. The proposed LSTM-AE-XGB model achieves the best overall performance across different classes, demonstrating its capacity to identify different traffic types. A precision, recall, and F1- score of 1.00 for “Normal” and “Neris” and up to 0.999 for the rest of the classes showcase its efficiency for identification of various traffic activities and its generalization ability to capture complex and unknown botnet patterns.

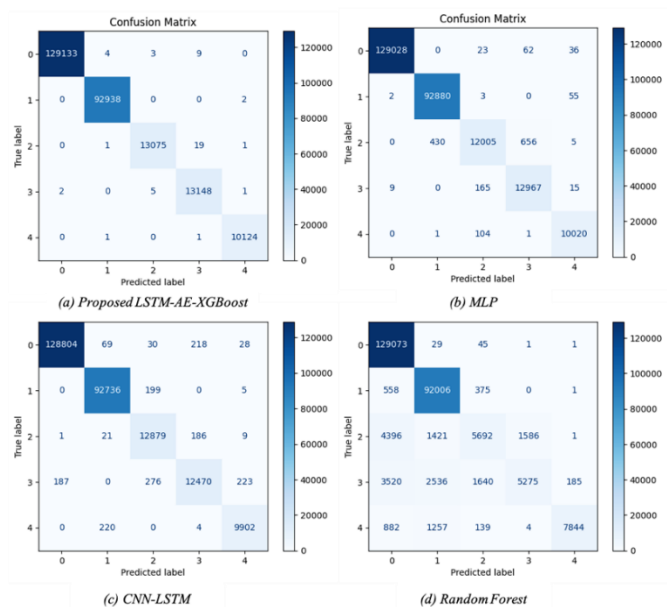


Figure 7. The confusion matrix resulting from the proposed model and other classifiers

5. CONCLUSIONS

In this study, we proposed a DL-based model to effectively analyse and detect botnet activities from IoT network traffic data captured by three parallel sensors. Long-short-term memory-based autoencoders are used for simultaneous characteristics. Traffic type is identified through a multiclass classification using the XGboost classifier over the NCC-2 dataset. The obtained results indicate that the proposed model is able to accurately detect botnet activity by achieving accuracy and precision up to 0.999. A comparative analysis with state-of-the-art ML and DL techniques for both feature extraction and classification demonstrates that the proposed LSTM-AE-XGBoost model presents a superior generalization ability.

In the future direction, the proposed model will be evaluated for detecting unknown botnet attacks and other cyber-attack patterns. Additionally, we envisage examining the proposed LSTM-AE-XGB-based IDS deployment into real-time IoT network.

REFERENCES

- [1] Vu, L., Nguyen, Q.U., Nguyen, D.N., Hoang, D.T., Dutkiewicz, E. (2020). Learning latent representation for IoT anomaly detection. *IEEE Transactions on Cybernetics*, 52(5): 3769-3782. <https://doi.org/10.1109/tcyb.2020.3013416>
- [2] Haseeb, J., Mansoori, M., Hirose, Y., Al-Sahaf, H., Welch, I. (2022). Autoencoder-based feature construction for IoT attacks clustering. *Future Generation Computer Systems*, 127: 487-502. <https://doi.org/10.1016/j.future.2021.09.025>
- [3] Velasco-Mata, J., González-Castro, V., Fidalgo, E., Alegre, E. (2023). Real-time botnet detection on large network bandwidths using machine learning. *Scientific Reports*, 13(1): 4282. <https://doi.org/10.1038/s41598-023-31260-0>
- [4] Nguyen, T.N., Ngo, Q.D., Nguyen, H.T., Nguyen, G.L. (2022). An advanced computing approach for IoT-botnet detection in industrial Internet of Things. *IEEE Transactions on Industrial Informatics*, 18(11): 8298-8306. <https://doi.org/10.1109/tii.2022.3152814>
- [5] Sattari, F., Farooqi, A. H., Qadir, Z., Raza, B., Nazari, H., Almutiry, M. (2022). A hybrid deep learning approach for bottleneck detection in IoT. *IEEE Access*, 10: 77039-77053. <https://doi.org/10.1109/access.2022.3188635>
- [6] Szykiewicz, P. (2022). Signature-Based Detection of Botnet DDoS Attacks. In: Kołodziej, J., Repetto, M., Duzha, A. (eds) *Cybersecurity of Digital Service Chains*. Lecture Notes in Computer Science, vol 13300. Springer, Cham, pp. 120-135. https://doi.org/10.1007/978-3-031-04036-8_6
- [7] Saba, T., Rehman, A., Sadad, T., Kolivand, H., Bahaj, S. A. (2022). Anomaly-based intrusion detection system for IoT networks through deep learning model. *Computers and Electrical Engineering*, 99: 107810. <https://doi.org/10.1016/j.compeleceng.2022.107810>
- [8] Sharma, B., Sharma, L., Lal, C., Roy, S. (2023). Anomaly based network intrusion detection for IoT attacks using deep learning technique. *Computers and Electrical Engineering*, 107: 108626. <https://doi.org/10.1016/j.compeleceng.2023.108626>
- [9] Ngo, D.M., Lightbody, D., Temko, A., Pham-Quoc, C., Tran, N.T., Murphy, C.C., Popovici, E. (2023). Network attack detection on IoT devices using 2D-CNN models. In *International Conference on Intelligence of Things*, pp. 237-247. https://doi.org/10.1007/978-3-031-46749-3_23
- [10] Al-Fawa'reh, M., Al-Fayoumi, M., Nashwan, S., Fraihat, S. (2022). Cyber threat intelligence using PCA-DNN model to detect abnormal network behavior. *Egyptian Informatics Journal*, 23(2): 173-185. <https://doi.org/10.1016/j.eij.2021.12.001>
- [11] Alzahrani, M.Y., Bamhdi, A.M. (2022). Hybrid deep-learning model to detect botnet attacks over internet of things environments. *Soft Computing*, 26(16): 7721-7735. <https://doi.org/10.1007/s00500-022-06750-4>
- [12] Anande, T.J., Leeson, M.S. (2023). Synthetic network traffic data generation and classification of advanced persistent threat samples: A case study with GANs and XGBoost. In *International Conference on Deep Learning Theory and Applications*, pp. 1-18. https://doi.org/10.1007/978-3-031-39059-3_1
- [13] Yin, Y., Jang-Jaccard, J., Xu, W., Singh, A., Zhu, J., Sabrina, F., Kwak, J. (2023). IGRF-RFE: a hybrid feature

- selection method for MLP-based network intrusion detection on UNSW-NB15 dataset. *Journal of Big Data*, 10(1): 1-26. <https://doi.org/10.1186/s40537-023-00694-8>
- [14] Hoang, X.D., Vu, X.H. (2022). An improved model for detecting DGA botnets using random forest algorithm. *Information Security Journal: A Global Perspective*, 31(4): 441-450. <https://doi.org/10.1080/19393555.2021.1934198>
- [15] Putra, M.A.R., Hostiadi, D.P., Ahmad, T. (2022). Botnet dataset with simultaneous attack activity. *Data in Brief*, 45: 108628. <https://doi.org/10.1016/j.dib.2022.108628>
- [16] Hosseini, S., Nezhad, A.E., Seilani, H. (2022). Botnet detection using negative selection algorithm, convolution neural network and classification methods. *Evolving Systems*, 13(1): 101-115. <https://doi.org/10.1007/s12530-020-09362-1>
- [17] Mohiuddin, G., Lin, Z., Zheng, J., Wu, J., Li, W., Fang, Y., Wang, S., Chen, J., Zeng, X. (2023). Intrusion detection using hybridized meta-heuristic techniques with Weighted XGBoost Classifier. *Expert Systems with Applications*, 232: 120596. <https://doi.org/10.1016/j.eswa.2023.120596>
- [18] da Cruz, M.A., Abbade, L.R., Lorenz, P., Mafra, S.B., Rodrigues, J.J. (2022). Detecting compromised IOT devices through XGBoost. *IEEE Transactions on Intelligent Transportation Systems*, 24(12): 15392-15399. <https://doi.org/10.1109/tits.2022.3187252>
- [19] Li, R., Li, Q., Zhou, J., Jiang, Y. (2021). ADRIoT: an edge-assisted anomaly detection framework against IoT-based network attacks. *IEEE Internet of Things Journal*, 9(13): 10576-10587. <https://doi.org/10.1109/jiot.2021.3122148>
- [20] Andresini, G., Appice, A., Malerba, D. (2021). Autoencoder-based deep metric learning for network intrusion detection. *Information Sciences*, 569: 706-727. <https://doi.org/10.1016/j.ins.2021.05.016>
- [21] Lahasan, B., Samma, H. (2022). Optimized deep autoencoder model for internet of things intruder detection. *IEEE Access*, 10: 8434-8448. <https://doi.org/10.1016/j.comnet.2021.108399>
- [22] Bårli, E.M., Yazidi, A., Viedma, E.H., Haugerud, H. (2021). DoS and DDoS mitigation using variational autoencoders. *Computer Networks*, 199: 108399. <https://doi.org/10.1016/j.comnet.2021.108399>
- [23] Vu, L., Nguyen, Q.U., Nguyen, D.N., Hoang, D.T., Dutkiewicz, E. (2020). Learning latent representation for IoT anomaly detection. *IEEE Transactions on Cybernetics*, 52(5): 3769-3782. <https://doi.org/10.1109/tcyb.2020.3013416>
- [24] Thakur, S., Chakraborty, A., De, R., Kumar, N., Sarkar, R. (2021). Intrusion detection in cyber-physical systems using a generic and domain specific deep autoencoder model. *Computers & Electrical Engineering*, 91: 107044. <https://doi.org/10.1016/j.compeleceng.2021.107044>
- [25] Khan, S.S., Mailewa, A.B. (2023). Detecting network transmission anomalies using autoencoders-SVM neural network on multi-class NSL-KDD dataset. In 2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, pp. 0835-0843. <https://doi.org/10.1109/ccwc57344.2023.10099056>
- [26] Kalakoti, R., Nömm, S., Bahsi, H. (2022). In-depth feature selection for the statistical machine learning-based botnet detection in IoT networks. *IEEE Access*, 10: 94518-94535. <https://doi.org/10.1109/access.2022.3204001>
- [27] Garcia, S., Grill, M., Stiborek, J., Zunino, A. (2014). An empirical comparison of botnet detection methods. *Computers & Security*, 45: 100-123. <https://doi.org/10.1016/j.cose.2014.05.011>
- [28] Hostiadi, D.P., Ahmad, T. (2021). Dataset for Botnet group activity with adaptive generator. *Data in Brief*, 38: 107334. <https://doi.org/10.1016/j.dib.2021.107334>
- [29] Que, Z., Liu, Y., Guo, C., Niu, X., Zhu, Y., Luk, W. (2019). Real-time anomaly detection for flight testing using AutoEncoder and LSTM. In 2019 International Conference on Field-Programmable Technology (ICFPT), Tianjin, China, pp. 379-382. <https://doi.org/10.1109/icfpt47387.2019.00072>
- [30] Said Elsayed, M., Le-Khac, N.A., Dev, S., Jurcut, A.D. (2020). Network anomaly detection using LSTM based autoencoder. In Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks, pp. 37-45. <https://doi.org/10.1145/3416013.3426457>
- [31] Maleki, S., Maleki, S., Jennings, N.R. (2021). Unsupervised anomaly detection with LSTM autoencoders using statistical data-filtering. *Applied Soft Computing*, 108: 107443. <https://doi.org/10.1016/j.asoc.2021.107443>
- [32] Le, T.T.H., Oktian, Y.E., Kim, H. (2022). XGBoost for imbalanced multiclass classification-based industrial internet of things intrusion detection systems. *Sustainability*, 14(14): 8707. <https://doi.org/10.3390/su14148707>