DOI: http://doi.org/10.32792/utq.jceps.10.01.01

A Comparative Study: Effect of Feature Selection Methods on Detecting Botnet Attacks in IoT Devices by Using Deep Learning

Zainab Mohammed Ghadhban [1] Hyder Yahya Alshaeaa [2]

msc21co12@utq.edu.iq haideryhya.comp@utq.edu.iq

[1, 2,] Department of Computer Science, College of Computer Science and Mathematics, University of Thi-Qar, Iraq

Received 28/4/2023, Accepted 29/5/2023, Published 11/6/2023

Abstract:

The Internet of Things, or IoT, is now an important technology that is the basis for various innovations in intelligent environments, including smart homes and innovative healthcare. Due to their architecture, many IoT devices suffer from security issues, leading to increased electronic threats targeting IoT devices, facilitating abuse and lack of security control. The most common attack in IoT environments is the botnet attack, which supports various criminal activities. In this study, This article aims to study different methods for selecting features and comparing them to find the best possible strategies for selecting and reducing features to detect botnet attacks in IoT devices. By using the UNSW-NB15 dataset, we will analyze the system's performance suggested to solve the classification issue. On the UNSW-NB15 dataset, the results obtained using the LSTM, BRNN, and GRU classifiers were analyzed to solve the binary classification issue and multiclass classification issues and compare the performance of three different selection features (Correlation method, GNDO method, and Lasso method) of network intrusion detection. The proposed system was evaluated by using different performance metrics and comparing the various techniques to show better performance. The results showed with that a Filter method (Correlation) for selecting features is better than other methods, and the model GRU in deep learning got the highest accuracy, amounting to 92.71% and 78.62% for both binary and multiple classifications, respectively. This study can potentially be applied in practical settings to detect real-time network intrusions with a dynamic nature.

Keywords: IOT, botnet attack, deep learning, select features methods.

Introduction:

The Internet of Things (IoT) has recently gained prominence in academia and business. The IoT has become an important technology that is the basis for various innovations in intelligent environments, including smart homes, innovative healthcare, and brilliant everything. In addition, the IoT has been adopted in multiple applications to improve services due to the rapid growth of IoT devices and technological progress [1]. The phrase "Internet of Things" was first used by Kevin Ashton in 1999 [2] to refer to sensor technology's limitless data collection possibilities.The IoT is more than 20 times more scalable than intelligent devices compared to current Information technology(IT) roles [3]. An increase in IoT adoption is expected across various industries, including utilities, healthcare, government, physical security, and automobiles.Based on this, electronic threats targeting IoT devices have increased because most IoT devices are connected to the Internet, facilitating abuse and lack of security control.The need for appropriate security measures implemented by IoT manufacturers [4].

IoT devices face many difficulties as technology evolves Due to their architecture, many IoT devices suffer from security issues, making them vulnerable to various attacks [5].For example, networked and compromised IoT devices are used in IoT botnet attacks, which the attacker uses to launch a DDoS attack, where the attack attempts to flood incoming traffic from many sources simultaneously. The most common attack is the Mirai malware, born in 2016. The attack process consists of several stages where vulnerable devices join the botnet and are under the control of the attacker's command and control center [6]. Botnets support various criminal activities, including click fraud, phishing, the distribution of malware, spam emails, and the unauthorized exchange of information or material in the Internet of Things. IoT device lack of security software, use of insecure development practices, mismanagement of security, and cybersecurity awareness issues are all factors that contribute to attacks. As the number of IoT devices increases globally, so does the sophistication and sophistication of IoT botnet attacks. Several solutions, including the intrusion detection system, have been developed to detect botnet attacks. It makes it possible to monitor the network and identify suspicious network activity. An intrusion detection system uses various technologies to detect malicious network activity. This technology is advancing significantly when machine and deep learning methods can be used to identify botnets because of their ability to identify malicious traffic and protect against botnet attacks. More cyber-attacks can compromise IoT devices due to the increased interest in these devices. Intrusion detection systems have been appropriately used as a solution to reduce botnet attacks.

IoT device manufacturers failed to implement sufficient security controls to protect devices from remote attacks, according to the 2019 SonicWall Cyber Threat Report [7], which led to a 217.5% increase in IoT attacks in 2018. Successful application of machine and deep learning confirms learning technologies in many areas have been used, which has led to the rise in the potential solutions that machine learning can provide in the security sector in recent years[3]. Investment in artificial intelligence, big data, and analytics will increase due to using machine learning for security purposes. Big tech companies have already used machine learning to detect threats against endpoint devices. Unlike traditional network intrusion detection systems, which are usually signature-based and used only to identify known attacks, they cannot detect zero-day vulnerabilities, making it difficult to determine whether network traffic is malicious. At the same time, deep learning and machine learning-based detection system are used as effective detection mechanisms to distinguish the variances of attacks and determine the nature of the traffic pattern [8].

The researchers used a feature selection technique to capture the botnet's attack pattern to speed up botnet detection. A different set of features extracted from the network traffic were used to identify botnets in the early stage. They identify malware using several features. Having more features does not significantly affect the detection of command and control traffic because a high level of detection performance is reached when the number of components is increased [9]. Dimensionality reduction can thus enhance the accuracy of deep learning and machine learning models. Because it enables both the detection and prevention of attacks through appropriate countermeasures, early detection of botnet attacks is crucial. Malware class models can enhance botnet detection methods in the medium-sized IoT environment, making it easier to identify IoT botnets. Therefore, this article will focus on studying different methods for selecting features and comparing them to find the best possible strategies for selecting and reducing features to detect botnet attacks in IoT devices.

1. RELATED WORK

The application of deep learning and machine learning models in detecting various attacks is increasing due to their high computational efficiency and accuracy. The following section presents a literature review and some relevant studies of IoT botnet attack detection methods ranging from deep learning to machine learning.

Wan et al. (2020) introduced technology to discover potential botnets by examining network activity patterns in network packets. They identified and categorized the threat posed by botnets by evaluating these characteristics using the proposed deep learning algorithms. LSTM is superior to RNN in these criteria. LSTM was able to provide superior results with the help of RNN. As a result, the combination of LSTM and RNN can be an effective model for identifying botnets[14][10].

Nugraha et al. (2020), the CTU-13 botnet traffic dataset was used to test simulations of four distinct deep learning models: Convolutional Neural Network (CNN), Long-Short Term Memory (LSTM), Hybrid CNN-LSTM, and Multi-layer Perception (MLP) for detecting botnets. The results showed that their deep learning models outperformed other deep learning models in their ability to accurately and reliably detect known and unknown bot traffic [15][11].

Segun et al. (2021) developed Federated Deep Learning (FDL) technology to detect zero-day bot attacks and prevent data leaks in edge IoT devices. This method classifies network traffic using an optimized Deep Neural Network (DNN) architecture. The results showed the ability of the FDL model to: detect zero-bot attacks with a high classification performance; integrate data privacy and security; withstand low communication loads; need little memory to store the training data; and it has low network latency. So, the FDL approach performed better than the DL and distributed DL methods [11][12].

Alkahtani et al. (2021) proposed a hybrid deep learning algorithm based on Convolutional Neural Network and Long Short-Term Memory (CNN-LSTM) to detect Bashlite and Mirai botnet attacks(is malware that infects Linux systems to launch distributed denial-of-service attacks (DDoS)) on nine commercial IoT devices. The experimental results showed that the CNN-LSTM model was superior in detecting bot attacks from doorbells with 90.88% and 88.61% accuracy, respectively. The system achieved good accuracy (88.53%) in recognizing attacks from thermostats. Regarding accuracy metrics, the system detected bot attacks from security cameras with an accuracy of 87.19%, 89.23%, 87.76%, and 89.64%. The CNN-LSTM model generally detects bot attacks from various IoT devices with high accuracy [12][13].

Idrissi et al. (2021) introduced a solution based on deep learning of convolutional neural networksbot intrusion detection systems called BotIDS (CNN). Their goal is to design, build, and test their IDS using a specialized Bot-IoT dataset against some well-known botnet attacks. Their results from BotIDS are encouraging compared to deep learning technologies such as RNN, LSTM, and GRU [13][14].

Hasan et al. (2022) proposed a hybrid and intelligent defense system that supports deep learning to protect the IoT infrastructure from bot attacks. The suggested mechanism has undergone a thorough evaluation using the most recent dataset, traditional and extended performance evaluation measures, and leading-edge DL benchmark techniques. Accurate recognition of complex, multivariate bot attacks is where the proposed methods are superior. The speed efficiency of their proposed method also showed excellent results [10][15].

2. RESEARCH DESIGN AND RESEACH METHODOLOGY

The proposed approach divided network traffic into offensive and benign categories to identify botnet traffic. This article examines how removing noisy features to preserve important properties affects the classification of legitimate and malicious events. The proposed methodology is shown in Figure 1. In the first stage, the database (UNSW-NB15) is used, then pre-processing as data cleansing, data preparation, and feature selection. It includes feature selection (filter, wrapper, and embed methods). Finally, the data from these three methods are classified into machine learning and deep learning. Accordingly, the data is classified as binary (benign data, attack data). And the second time, classify the data based on multiple classifications (benign data, attack data (i.e., nine attacks))

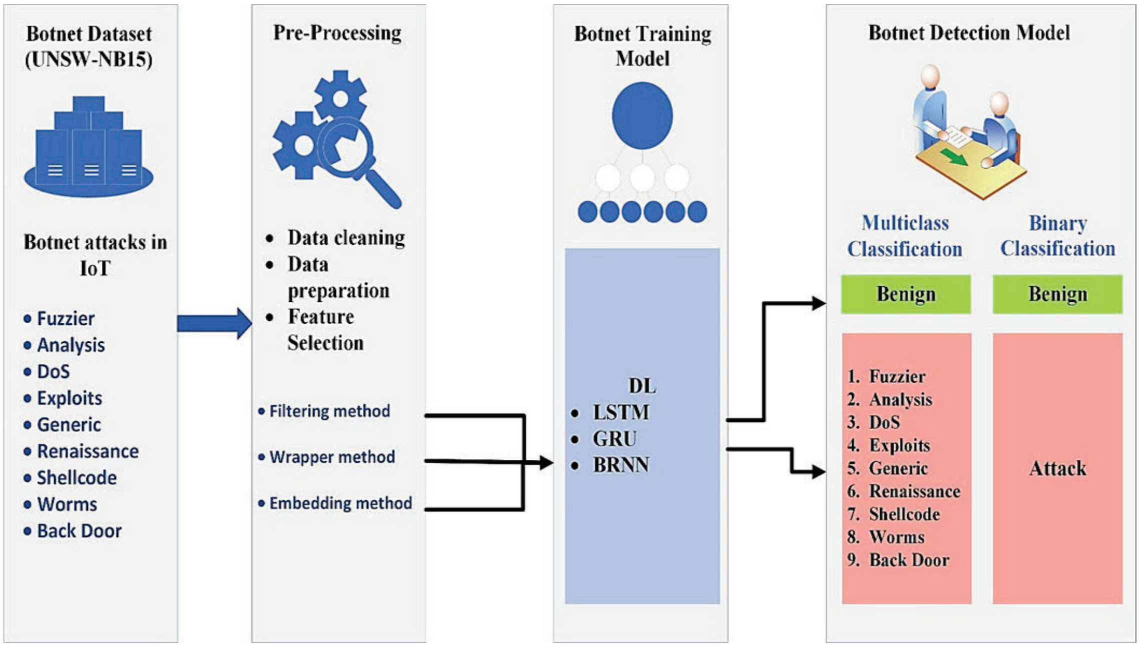
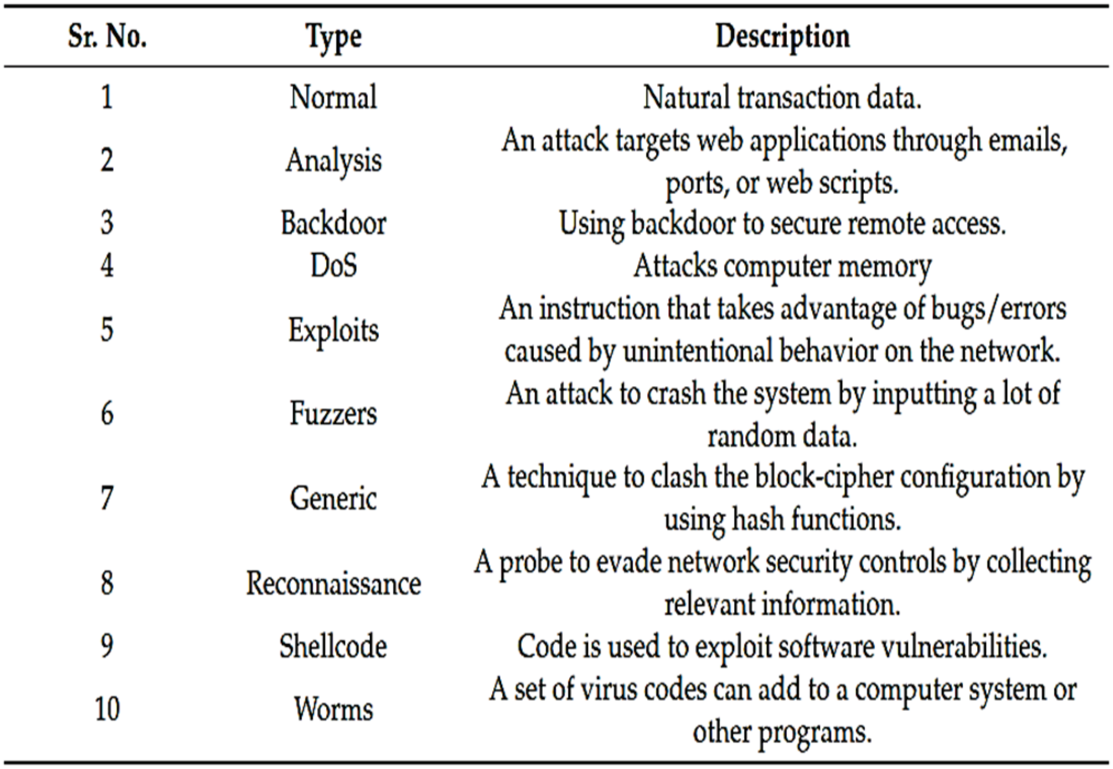


Figure 1. The proposed framework for detection Botnet attack in IoT using deep learning

a. Dataset Description

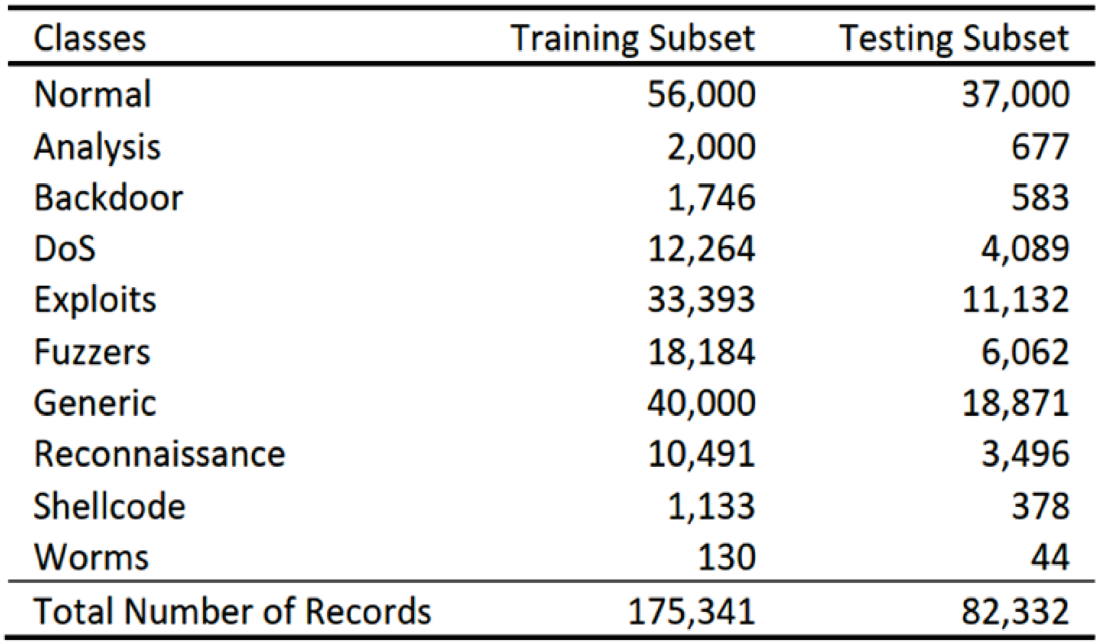
The University of New South Wales (UNSW) Cyber Range Lab, Canberra, used IXIA PerfectStorm software to integrate everyday activities with contemporary synthetic threats [16]. 100 GB of raw data was recorded with tcpdump. This dataset contains several injection attacks, including DoS, Fuzzers, Worms, Analysis, Exploits, Backdoors, General, Reconnaissance, and Shellcode. 11.6% were selected as the sample (257,673 records) of the original 2,218,761 records, with 175,341 records forming the training subset and 82,332 records from the test subset. This dataset contains 49 features. Table 1 describes the attack data set [17]. Table 2 shows the size of the logs in the training and test subsets for each attack [18].

Table 1. All attack datasets, 9 attacks and a normal case [17].



Sr. No.	Type	Description
1	Normal	Natural transaction data.
2	Analysis	An attack targets web applications through emails, ports, or web scripts.
3	Backdoor	Using backdoor to secure remote access.
4	DoS	Attacks computer memory
5	Exploits	An instruction that takes advantage of bugs/errors caused by unintentional behavior on the network.
6	Fuzzers	An attack to crash the system by inputting a lot of random data.
7	Generic	A technique to clash the block-cipher configuration by using hash functions.
8	Reconnaissance	A probe to evade network security controls by collecting relevant information.
9	Shellcode	Code is used to exploit software vulnerabilities.
10	Worms	A set of virus codes can add to a computer system or other programs.

Table 2. Number of records in training and testing subsets for each attack [18].



Classes	Training Subset	Testing Subset
Normal	56,000	37,000
Analysis	2,000	677
Backdoor	1,746	583
DoS	12,264	4,089
Exploits	33,393	11,132
Fuzzers	18,184	6,062
Generic	40,000	18,871
Reconnaissance	10,491	3,496
Shellcode	1,133	378
Worms	130	44
Total Number of Records	175,341	82,332

2.2 Preprocessing

2.2.1. Data Cleaning

The labels class does not need all 49 features. One of the features, the total duration (dur), determined from the difference of values for these two input features, two input features, namely

record_start_time, and record_last_time, are redundant. Certain features, such as Source IP address, Source port number, Destination IP address, and Destination port number, only apply to the computer infrastructure and donot provide information necessary for intrusion detection. As a result, we remove these features while keeping the defining features from another input.
The attack_cat feature is of type nominal and contains the names of attack categories. For visualization, these features for Binary and Multi-class classification. Two of the 44 remaining input features (attack_cat and label) are class features. The names of attack categories are contained in the nominal type feature called "attack_cat" These features are necessary for binary and multi-class classification

2.2.2. Data Preparation

Some of the features, such as the category features "proto", "service", and "state" must be converted into numerical values for the classifiers to be able to detect the attacks.

2.3. Feature Selection

Feature selection is essential in solving a classification problem using machine and deep learning. The main issue of dimensionality is that irrelevant features may affect classification accuracy. Due to the poor model quality caused by exterior features, these factors make it challenging for cyber security experts to interpret traffic. As a result, they cannot take timely action to respond with appropriate incident-handling measures. It is essential to choose the right set of features. To do that, feature selection methods will select the best features that match their class labels. This is the ultimate goal of every strategy for feature selection. Overall, feature selection is required to identify the most pertinent features, omit features that are irrelevant to classification or eliminate noise features, and decrease the model's dimensionality. A few feature selection methods learn how each part affects the machine learning model and compute results. Feature selection includes filter, wrapper, and embedded techniques

2.3.1. Filter Method

This technique picks out a subset of features. The foundation of the filter method is a statistical method like the distance between classes. The dataset ranks feature scores; after that, to filter the dataset, irrelevant characteristics are employed. Because it only considers one variable at a time, this method is univariate and does not consider how variables are correlated. These include the chi-square, Gini index, entropy, information gain, Fisher score, correlation, and relief methods. We selected correlation in our proposal.

2.3.1.1 Correlation Method

Correlation coefficients calculate the inter-correlations between the features and the correlation between a subset of attributes and class. The correlation between a feature set and a class increases its relevance while increasing inter-correlation causes it to decline. CFS is frequently combined with other search techniques to select the best feature subset, including forward selection, backward elimination, bi-directional search, best-first search, and genetic search. The provided equation is for the CFS.

$$\boldsymbol{r}_{\boldsymbol{zc}=}\frac{\boldsymbol{k}_{\boldsymbol{rzi}}}{\sqrt{\boldsymbol{k}+\boldsymbol{k}(\boldsymbol{k}-\mathbf{1})\boldsymbol{r}_{\boldsymbol{ii}}}} \qquad (\mathbf{1})$$

Where (r_{zi}) is the average of the correlations between the subset features, and the class variable, it is the average inter-correlation between subset features, k is the number of subset features, and (r_{zi}) is the correlation between the summed feature subsets and the class variable [19].

2.3.2 Wrapper Method

The technique selects features that have been combined, assessed, and then compared to other combinations, iteratively improving a current set of features. A machine learning model must evaluate the integrated features and pertinent scores assignment. After that can evaluate the model's performance. The computation might be challenging in a variety of ways. One of the methods uses heuristics, such as forward and backward passes that add up or remove features. Another approach uses the best-first principle, for instance, the random hill-climbing algorithm. There is also a recursive feature elimination algorithm. Filtering approaches might not be as precise as the wrapped approach. However, it calls for computationally expensive data manipulation techniques

2.3.2.1. Generalized Normal Distribution Optimization (GNDO)

Generalized standard distribution optimization (GNDO) has recently been used to tackle nonlinear optimization issues. The theory of normal distribution served as the basis for this algorithm based on two main stages of optimization techniques: exploration and exploitation. These two GNDO stages are thoroughly explained in the remaining paragraphs.

- **Localized Exploitation**

The local exploitation stage looks at the mean of three chosen solutions: the best-so-far solution X, the position vector of the itch solution Xi^t, and the mean M of the keys (3). The promising region in the current generation t uses Eq. (4). Then, using a step size determined by Eq. (5), it performs a search around this promising region to produce a new trial solution that might be superior to the current one

$$Ti^t = \mu i + \delta i \times \eta, \quad \forall\, i = 1{:}\, N \tag{2}$$

$$\mu i = \frac{Xi^t + X^* + M}{3.0} \tag{3}$$

$$M = \frac{\sum_{i=1}^{N} x\, i^t}{N} \tag{4}$$

$$\delta i = \sqrt{1/3 + [(xi^T - \mu)^2 + (x^* - \mu)^2 + (M - \mu)} \tag{5}$$

WhereTi^t is the trail vector of the ith person at time t, μi is the ith person's average position, δi is the ith person's standard deviation, and M is the average position of the current population η is the penalty factor.

- **Global Exploration**

The optimization issue search space will extensively study in the global research phase to determine the optimal solution. This stage's mathematical formula is (6).

$$Ti^t = Xi\text{^}t + \beta \times (|\beth_3 \vdash| \times V1) + (1 - \beta) \times (\beth_4| \times V2) \tag{6}$$

where $|\beth_3$ *and* $\beth_4|$ *are two random numbers with a standard normal distribution,* β *is a random number between 0 and 1, is called the change parameter, and V1 and V2 are two trail vectors.*

2.3.3 Embedded Method

The embedded method examines and rates numerous training iterations based on each feature's importance. During model construction, it chooses features. A typical embedded feature selection method is the regularization technique Regularization adds more factors to an accurate machine-learning model to make it more real. Examples include Ridge Regression, and Lasso.

2.3.3.1 Lasso Method

The lasso was invented by Robert Tepcherani in 1996 [20]. Effectively choose and organize features. The Lasso approach limits the sum of the absolute values of the model parameters to a predetermined upper constraint. Deflation (regulation) punishes variable regression coefficients, reducing some to zero. During feature selection, variables with non-zero coefficients are included after shrinkage. This method reduces prediction error. In practice, the parameter for adjusting the severity of punishment is essential. When high enough, the operands should equal 0, reducing the dimensions. As the parameters grow, more coefficients become zero. Because the coefficients are reduced and omitted, the Lasso approach may accurately predict the results. Since the adjustment parameter increases bias and decreases variance, this is particularly useful when there is data and must determine the trade-off between bias and conflict. By omitting unnecessary variables, Lasso reduces overfitting and improves the interpretability of the model

2.4. Experimental Evaluation

In this Section, we determine which model is best for IoT botnet detection was determined by investigating performance differences type and DL model types. We first create an IoT botnet detection model based on the proposed framework. Both multiclass classification and binary classification. Are made Multiclass classification begins to classify benign and even other types of attacks, whereas binary classification classifies datasets as benign or attacked. We next use the testing sets to verify our DL models

2.4.1. Binary Classification

Nine attacks on IoT devices are treated as one in the binary classification model. Additionally, it makes a distinction between attacks and benign cases. Based on the DL models, we train our model utilizing the dataset amassed from each device. In deep learning, we are using Long Short-Term Memory Networks (LSTMs), Bidirectional recurrent neural networks (BRNN), and Gated Recurrent Unit (GRU) as a method of classification.

2.4.2. Multiclass Classification

Nine attacks are viewed as separate attacks in the multiclass classification model. We are using LSTM, BRNN, and GRU as a classification method in deep learning.

2.5. Performance Measurements

To determine how successful the recommended algorithms are in identifying botnet attacks, the measurement of the accuracy is one of the evaluation metrics, together with the recall, precision, and F1-score. The formulae for the parameters under consideration are as follows:

Recall: is calculated by dividing the total number of positives by the number of true positives.

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

Precision: is calculated by dividing the number of actual positive results by the number of anticipated positive results

$$Precision = \frac{TP}{TP + FP} \quad (8)$$

The F-score: is the harmonic mean of recall and accuracy.

$$F1 - score \ = \frac{2 * (Precision * Recall)}{(Precision * Recall)} \quad (9)$$

Accuracy: Accuracy is determined by measuring the actual value from the measured value.

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} \quad (10)$$

Where TP is true positive, FP is false positive , FN is false negative ,TN is true negative .

3. RESULTS OF THE EXPERIMENtT AND DISCUSSION

Botnets during a literature survey. The proposed technique needs to be improved in terms of scalability, accuracy, complex data, slow results, etc., for large datasets. A more feasible approach to address all these challenges needs to be considered. Deep learning is one such approach that combines many techniques. Deep-learning algorithms were used in IoT botnet attacks. Using the(UNSW-NB15) dataset, we will analyze the system's performance we developed for solving the classification issue. On the UNSW-NB15 dataset, we analyze the results obtained using the LSTM, BRNN, and GRU classifiers to solve the binary classification issue and multiclass classification issue of network intrusion detection. A lot of the studies focus on traditional datasets. Our main focus is on the more complicated network dataset, for which we use fewer features, and we compare the performance of three different feature selection methods. To obtain the best possible comparison results, we unified the number of features used for each of the three methods (select features) to be 20 features for binary classification and 21 for multiple classifications. The purpose of this article is to correctly categorize standard network traffic, network threats, and suspicious network activity

4.1 Deep Learning Algorithms

This section presents the results of DL models (LSTM, BRNN, and GRU) for detecting botnet attacks in IoT environments. Two binary and multiclass classification experiments were performed on properties selected from the three methods of feature selection. Table 3 shows the results of the respective DL models. With the binary classification of the data, which included two categories of normal attacks or attacks, the GRU on the features that were taken from the filter method using (correlation) achieved high accuracy measures of 92.71% with less processing time.

Table 3. Binary classification by using deep learning methods

Feature Selection	**Performance Measurements**	**LSTM**	**GRU**	**BRNN**
Wrapper Method (GNDO) 19+1 (44) =20	**Accuracy**	0.9094	0.9174	**0.921645**
	Precision	0.8984	0.9118	0.9327
	F-score	0.9316	0.9372	0.9391
	Recall	0.9673	0.9641	0.9456
	Processing Time (s)	2.111344e+03	1.430297e+03	4.234375e+00
Filter method	**Accuracy**	**0.9268**	**0.9271**	0.921115

(Correlation) 19+1 (44) =20	**Precision**	0.9365	0.9343	0.9198
	F-score	0.9426	0.9424	0.9394
	Recall	0.9488	0.9507	0.9598
	Processing Time (s)	2.119047e+03	1.404188e+03	4.734375e+00
Embedded method (Lasso) 19+1 (44) =20	**Accuracy**	0.9212	0.9219	0.921399
	Precision	0.9282	0.9308	0.9326
	F-score	0.9391	0.9398	0.9391
	Recall	0.9502	0.9490	0.9457
	Processing Time (s)	2.168375e+03	2.044312e+03	4.406250e+00
All 42 42+1 (44) =43	**Accuracy**	0.9259	0.9256	0.916393
	Precision	0.9332	0.9293	0.9051
	F-score	0.9433	0.9440	0.9370
	Recall	0.9537	0.9591	0.9711
	Processing Time (s)	1.965156e+03	2.143078e+03	4.828125e+00

Table 4 presents the results of the multiclass classification, in which the GRU also achieved high accuracy metrics of 78.62%. In general, the filter (correlation) method is better than the other methods for selecting features in both classifications (binary, multilayer) in the deep learning model because it showed the highest accuracy and the lowest processing time relative to the (BRNN) method and slightly more than the processing time (LSTM) according to the (LSTM) method. Link while to wrapper method (GNDO) and embedding method (LASSO) better in terms of results.

Table 4. Multiclass classification by using deep learning methods

Feature Selection	Performance Measurements	LSTM	GRU	BRNN
Wrapper Method (GNDO) 20+1 (43) =21	Accuracy	0.7761	0.7738	0.772671
	Precision	0.45748	0.42377	0.42431
	F-score	0.37312	0.36532	0.36968
	Recall	0.38923	0.38253	0.38955
	Processing Time (s)	1.997781e+03	2.363234e+03	4.562500e+00
Filter method (Correlation) 20+1 (43) =21	Accuracy	0.7836	0.7862	0.779773
	Precision	0.48993	0.46488	0.43884
	F-score	0.37262	0.37254	0.36895
	Recall	0.38871	0.39106	0.38725
	Processing Time (s)	1.968625e+03	2.347484e+03	5.375000e+00
Embedded method (Lasso) 20+1 (43) =21	Accuracy	0.7745	0.7718	0.773667
	Precision	0.39011	0.45768	0.51563
	F-score	0.37171	0.37718	0.38426
	Recall	0.38685	0.3843	0.39212
	Processing Time (s)	2.373312e+03	1.615938e+03	6.953125e+00
All 42 42+1 (43) =43	Accuracy	0.7806	0.7862	0.770873
	Precision	0.51972	0.48824	0.46362
	F-score	0.37292	0.37238	0.38335
	Recall	0.3891	0.38871	0.38973
	Processing Time (s)	3.560375e+03	2.531234e+03	7.375000e+00

4. CONCLUSION

This article discussed the issues of detecting botnet attacks in the IoT environment. Furthermore, we proposed three methods for feature selection from all features of the UNSW-NB dataset 15. The proposed method first involves feature selection based on the role of three feature selection methods. We left out some features because they had a low impact on the final output label and were highly correlated with other input features. Because the model was trained and tested on the most discriminating variables, the proposed approach to identify botnet attacks has a high level of accuracy when applied to the UNSW-NB15 dataset. We evaluated the capabilities of our models based on the data set used, and the results showed consistent classification accuracy. We evaluated our proposed system using different performance metrics and compared different techniques to show better performance. The results obtained in this study showed that the filter method (correlation) for selecting features is better than other methods because its results were better than the rest of the models and the processing time was less. The deep learning GRU model obtained the highest accuracy of 92.71% and 78.62% for both binary and multiple classifications, respectively. Moreover, this study can be applied in practical settings to detect real-time network intrusions of a dynamic nature. A study can be conducted to select suitable classifiers to discover and evaluate their effectiveness.

REFERENCES

[1] **M. Wazzan and D. Algazzawi, O. Bamasaq, A. Albeshri, L.Cheng,** . "Internet of Things botnet detection approaches"in Analysis and recommendations for future research. Applied Sciences, 5713 11, December 2021.

[2] K .**Ashton,. "That 'internet of things' thing.** ***RFID journal*** **, 97-114 22,July 2009**.

[3] **J.Kim and M. Shim, S.Hong, Y. Shin, E.Choi,** " Intelligent detection of iot botnets using machine learning and deep learning". *Applied Sciences*, 7009 19 ,October, 2020 .

[4] **A.Cirne, and P.R. Sousa, J.S. Resende, L. Antunes,** "IoT security certifications"in Challenges and potential approaches".in *Computers & Security*, *116*, 102669 1,may 2022.

[5] **L.A.Tawalbeh, and F. Muheidat, M. Tawalbeh, , M. Quwaider, M**. (2020). IoT Privacy and security " in Challenges and solutions. *Applied Sciences*, *10*(12), 4102 12,October , 2020 .

[6] **N. Koroniotis, and `N. Moustafa, E. Sitnikova, J. Slay** , "Towards developing network forensic mechanism for botnet activities in the IoT based on machine learning techniques. In *Mobile Networks and Management, 9th International Conference, MONAMI 2017, Melbourne, Australia, December 13-15, 2017, Proceedings 9* (pp. 30-44). Springer International Publishing 2018 .

[7] **SonicWall cyber threat report.** https://www.sonicwall.com/2019-cyber-threat-report/ (accessed 27 February 2023).

[8] **A. Muhammad, and M. Asad, A.R Javed**, "Robust early stage botnet detection using machine learning". In *2020 International Conference on Cyber Warfare and Security (ICCWS)* (pp. 1-6). IEEE, October,2020.

[9] **M. .Lefoane and L. Ghafir, S. Kabir, I.U. Awan** "Machine learning for botnet detection"in An optimized feature selection approach. In *The 5th International Conference on Future Networks & Distributed Systems* (pp. 195-200) , December 2020.

[10] **W.C Shi, and H.M . Sun, "** DeepBot: a time-based botnet detection with deep learning" in Soft Computing, 24, 16605-16616, 2020 .

[11] **B. Nugraha, A. Nambiar and T. Bauschert, "**Performance Evaluation of Botnet Detection using Deep Learning Techniques," 2020 11th International Conference on Network of the Future (NoF), Bordeaux, France, pp. 141-149, doi: 10.1109/NoF50125.2020.9249198 ,2020**.**

[12] **S. I. Popoola, R. Ande, B. Adebisi, G. Gui, M. Hammoudeh and O. Jogunola**, "Federated Deep Learning for Zero-Day Botnet Attack Detection in IoT-Edge Devices," in *IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3930-3944, doi: 10.1109/JIOT.2021.3100755,1 March, 2022..

[13] **H. Alkahtani, and T.H. Aldhyani**, " Botnet attack detection by using CNN-LSTM model for Internet of Things applications" in *Security and Communication Networks*, 23 January 2021..

[14**] I. Idrissi, and Boukabous, M. Azizi, O. Moussaoui, H. El Fadili** " Toward a deep learning-based intrusion detection system for IoT against botnet attacks". *IAES International Journal of Artificial Intelligence*, 110,10, January 2021

[15] **T. Hasan et al.,** "Securing Industrial Internet of Things Against Botnet Attacks Using Hybrid Deep Learning Approach," in IEEE Transactions on Network Science and Engineering, doi: 10.1109/TNSE.,.3168533,2022.

[16] The UNSW-NB15 Dataset | UNSW Research, https://research.unsw.edu.au/projects/unsw-nb15-dataset ,2015

[17] **I. Ahmad, and Q.E.Ul Haq ,M. Imran, M.O Alassafi, R.A. AlGhamdi**.," An efficient network intrusion detection and classification system. *Mathematic"s*, 530,10,marach,2022.

[18] **Z. Zoghi, and G. Serpen,** . "Unsw-nb15 computer security dataset " in Analysis through visualization. *arXiv preprint arXiv:2101.05067,2021*

[19] **A.G. Karegowda and A.S. Manjunath, M.A. Jayaram**," Comparative study of attribute selection using gain ratio and correlation-based feature selection" in *International Journal of Information Technology and Knowledge Management*, 271-277, 2, February, 010.

.

[20] **P. Ghosh, and S. Azam, M. Jonkman, A.Karim,F.J.M. Shamrat, E. Ignatious**, F. De Boer," Efficient prediction of cardiovascular disease using machine learning algorithms with relief and LASSO feature selection techniques" in *IEEE Access*, 19304-19326, September 2021.