Section A-Research paper

A Reliable Technique for Spotting Botnet Attacks in Internet of Things Applications

Dr. Mohd Uruj Jaleel
Associate Professor
Department of Computer Science and Engineering
DR. K.N. Modi Institute of Engineering & Technology, Modinagar, Ghaziabad, Uttar Pradesh, India
ORCID ID: 0000-0002-2679-2531
Email: dr_urujjaleel@yahoo.com

Mr. Parbhat Gupta
Assistant Professor
Department of Computer Science and Engineering
DR. K.N. Modi Institute of Engineering & Technology, Modinagar, Ghaziabad, Uttar Pradesh, India
ORCID ID: 0009-0002-6100-6839
Email: parbhatgupta3@gmail.com

Mr. Ajit Singh
Assistant Professor
Department of Computer Science and Engineering
DR. K.N. Modi Institute of Engineering & Technology, Modinagar, Ghaziabad, Uttar Pradesh, India
Email: myemailajit@gmail.com

Dr. Payal Gulati
Assistant Professor
Department of Computer Engineering
J.C. Bose UST, YMCA

Mr. Ankur Biswas
Research Scholar
Department of Computer Science and Engineering
Adamas University, Kolkata, India
ORCID ID: 0000−0002−9554−219X
Email: ankur2u@gmail.com

Mr. Arun Soni
Certified Ethical Hacker (CEH)-EC Council, Digital Evidence Specialist (DES)
Asian School of Cyber Laws (ASCL), Pune, India
Email: info@arunsoni.in

ABSTRACT

A newly developed dataset is used for efficient feature selection and efficient Bot-IoT attack identification in an IoT network context. The dataset includes information about botnet attacks, regular traffic flows, the Internet of Things, and countless more cyber attacks. The realistic simulation is employed for the production of this dataset with effective information features in order to track the precise traffic. Similar to this, more features were extracted and integrated with the extracted features set to improve the performance of machine learning models and effective prediction models. The research

effort suggested employing deep learning algorithms like LSTM and CNN for effective botnet detection in IoT networks. By conducting in-depth tests with the most pertinent publically accessible dataset (Bot-IoT) in binary and multi-class classification scenarios, the efficacy of this strategy was confirmed. Because of its consistent updates, extensive attack diversity, and variety of network protocols, the Bot IoT was utilised as a dataset in this situation. We use the Bot-IoT dataset to assess our suggested strategy. Analysis of the simulation results revealed that, in comparison to the current BLSTM, our proposed method is effective and can, on average, produce better performance results.

Keywords: IOT, Botnet, Cyber, Attack, LSTM, CNN, Accuracy.

1. INTRODUCTION

An assault on a computer or computer network refers to any attempt to access or use a resource without authorization in order to discover, modify, impair, destroy, seize, or obtain data. Any antagonistic action that targets PC data frameworks, foundations, PC businesses, or PC gadgets is referred to as a "digital assault." An aggressor is a person or entity that tries to access data, functions, or another restricted area of the system without authorization and possibly with malicious intent. Cyber attacks may be crucial for digital combat or digital psychological warfare, depending on the circumstances. A digital assault may originate from an unidentified source and be used by sovereign states, individuals, groups, societies, or associations. Sometime referred to as a digital weapon, an object that functions with a digital attack [1].

The Internet of Things (IoT) describes the arrangement of physical objects that have been fitted with sensors, software, and other advancements to connect and exchange data with other devices and systems over the Web.

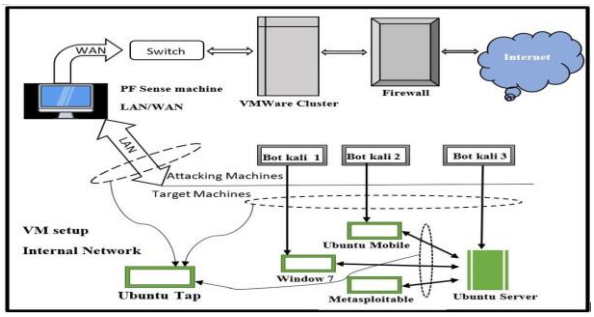

Figure 1: Bot-IOT

A freshly created dataset is used to identify the precise Bot-IoT assaults distinguishing proof in IoT network environment [3] [10]. The collection includes information about common traffic streams, the Web of Things, and a few other types of botnet attack traffic streams. The practical proving ground is used to strengthen this dataset with effective data highlights in order to follow the precise traffic and create a powerful dataset [4].

Today, Web of Things (IoT) innovation is growing step by step, and more and more devices are associated with this innovation. Using this innovation will make your daily life more convenient and efficient [5] [7] [9].

1.1 EXISTING SYSTEM

Deep learning (DL) is an effective strategy to detect botnet attacks. In any case, the amount of traffic information and required storage space for an organization is typically enormous. As such, implementing a DL strategy in memory-driven IoT devices is very difficult. This white paper massively reduces the dimensionality of the elements of IoT network traffic information. The framework has been enhanced by reducing component dimensionality. Then, at that point, perform quasi-breed deep learning computations such as LSTM and Auto-Encoder. It also uses a bidirectional LSTM. Comprehensive experiments are performed using the BoT-IoT dataset to confirm the feasibility of the proposed crossover DL technique. The results show that LAE radically reduces the expected storage

space for large-scale network traffic information capacity by 91.89%, outperforming state-of-the-art dimensionality reduction techniques by 18.92-27.03%. Despite the significantly reduced include size, the profound BLSTM model shows a strong effect on model under fitting and over fitting. It also achieves great throwing power in parallel and multiclass ordering situations.

Deep Learning (DL) has been generally proposed for botnet assault discovery in Web of Things (IoT) organizations [2].

Distinguishing botnet dangers has been a continuous exploration try. AI (ML) procedures have been broadly utilized for botnet location with stream based highlights [6] [14].

A botnet is a malware program that a programmer remotely controls called a botmaster. Botnet can perform huge digital assaults, for example, DDOS, SPAM, click-extortion, data, and personality taking [8].

Dispersed Forswearing of Administration (DDoS) assaults have caused huge disturbances in the activities of Web based administrations [11].

1.2 PROBLEM IDENTIFICATION

Tracking and analysing the attacks themselves, into which typical security solutions provide visibility, and identifying which attacks originated from botnets are more often used methods for identifying botnets. Many different cyber attacks can be carried out using botnets, which might include thousands of hosts. In particular, they can overload targets' networks and devices with traffic and steal data from hosts that have been infected with the bots.

After the literature survey some of the observation is carried out, which is as followings-

- Tracking and analysing the assaults themselves, into which typical security solutions provide visibility, and identifying which attacks came from botnets is a more popular technique for identifying botnets.
- Botnets may contain thousands of hosts and can be used to execute a variety of cyber-based attacks, in particular flooding target's networks and devices with too much traffic and stealing data from hosts infected with the bots.
- It doesn't efficient for large volume of data's and have some theoretical limits.

1.3 PROPOSED WORK

The main contribution of the proposed research work is as followings-

- Obtaining the Bot-IOT dataset from the Kaggle machine learning reciprocity.
- To use LSTM and CNN, two deep learning algorithms, to improve performance.
- To use dimensionality reduction techniques, like Principle Component Analysis (PCA), to cut down on the number of dimensions in the dataset.
- To put the automatic encoder into practice for compressing the raw data.
- To evaluate the performance metrics and improve.

Step-

- From the dataset repository, the Bot-IoT dataset will be used as input in this system.

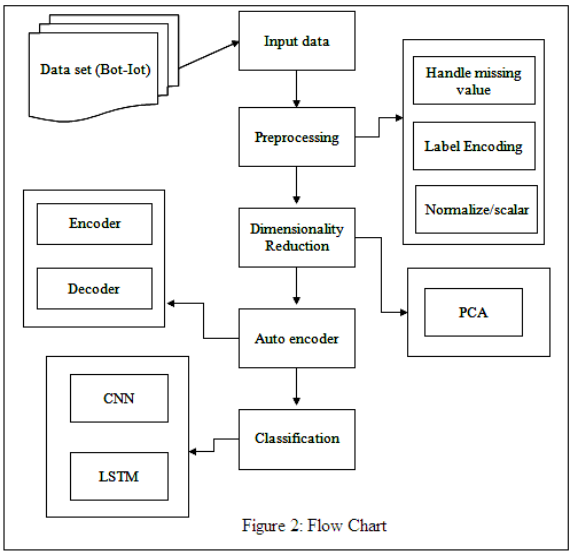

Figure 2: Flow Chart

- Next, we must put the data preprocessing phase into action. We must deal with the missing values in this stage to prevent incorrect prediction, encode the label for the input data, and normalize/scale the input data.
- The feature dimensionality reduction step, such as Principal Component Analysis, must then be implemented (PCA).
- Deep learning techniques like Long Short Term Memory (LSTM) and Convolutional Neural Network must be implemented (CNN).
- The simulation results also demonstrate that performance criteria like recall, accuracy, and precision will all improve.

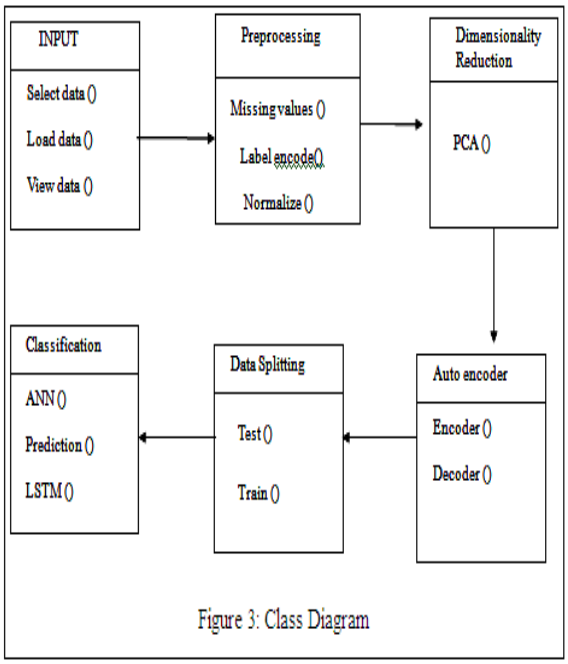

Figure 3: Class Diagram

Figure 3 is presenting the class diagram of the proposed model.

2. METHODOLOGY

In this system, the Bot-IoT dataset is taken as input. The input data is taken from the dataset repository. Then, we have to implement the data preprocessing step, next we have to handle the missing values for avoid wrong prediction, to encode the label for input data and normalize/scaling the input data. Then, we have to implement the feature dimensionality reduction such as Principal Component Analysis (PCA) is one of the common linear transformation methods while kernel methods, spectral methods and DL methods employ non-linear transformation techniques. Next, we have to implement the Auto encoder technique is an unsupervised DL method that produces latent-space representation of input data at the hidden layer. Different auto encoder architectures have been proposed to reduce the feature dimensionality in most popular network intrusion datasets. Then, we have to implement the deep learning algorithms such as Long Short term Memory (LSTM) and Convolution Neural Network (CNN). Finally, the simulation results shows that the performance metrics such as accuracy, precision, recall and confusion matrix. Such approach is efficient for large number of datasets.

The following phases serve as a foundation for the methodological step:

- Data selection
- Data preprocessing
- Data normalization
- Dimensionality reduction
- Auto encoder
- Classification
- Result Generation

3. Result Generation

On the basis of the overall categorization and forecast, the Final Result will be generated. The effectiveness of the suggested strategy is assessed using metrics such

- Accuracy
- ROC Accuracy
- Precision
- Recall
- F1-measure
- Sensitivity
- Specificity

Based on the total classification and projection, the final outcome will be determined. Create the confusion matrix before calculating the outcome parameters.

True Positive (TP): Predicted values correctly predicted as actual positive

False Positive (FP) : Predicted values incorrectly predicted an actual positive ie Negative values predicted as positive

False Negative (FN): Positive values predicted as negative

True Negative (**TN**): Predicted values correctly predicted as an actual negative

We compute the accuracy test from the confusion matrix:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

This framework shows the revised and wrong expectations, in correlation with the real marks. Every disarray network line shows the Real/Genuine marks in the test set, and the segments show the anticipated names by classifier. Something to be thankful for about the disarray grid is that it shows the model's capacity to effectively foresee or isolate the classes.

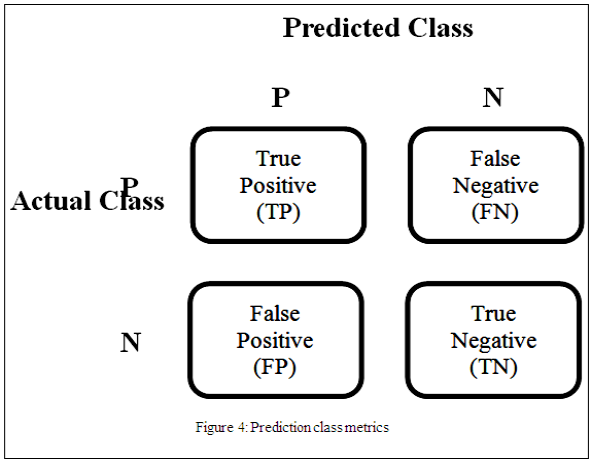

Figure 4: Prediction class metrics

3.1 RESULT AND ANALYSIS

The proposed calculation is carried out using Python version 3.7. The sklearn, numpy, pandas, matplotlib, pyplot, seaborn, and los libraries help us use the capabilities offered by Spyder climate for various tactics such as choice tree, arbitrary forests, and credulous bayes.

The dataset is taken to begin the simulation. PkSeqID, stime, flgs, flgs number, proto, proto number, saddr, sport daddr, dport pkts, bytes, statestate number, Itime, seq, dur etc are some of the properties mentioned in this dataset.

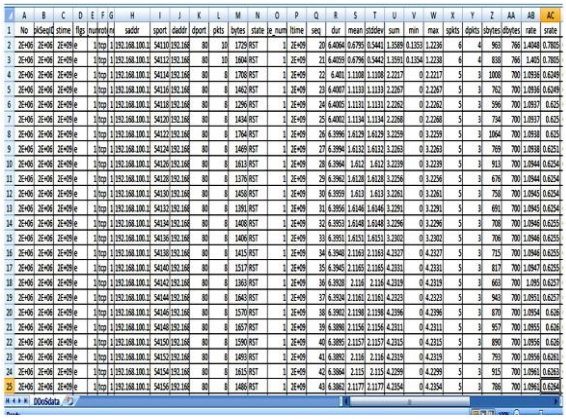

Figure 5.1: Original dataset in .csv fil

The Figure 5.1 is showing the dataset, which is taken from the kaggle machine learning website.

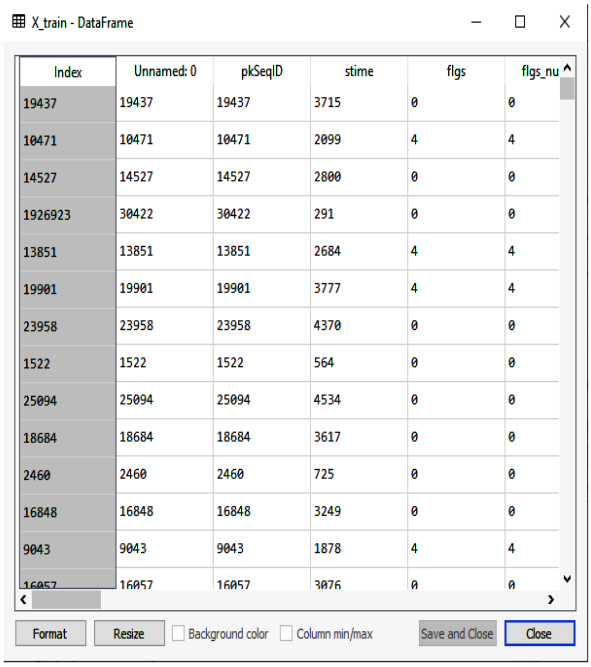

Figure 5.2: X train

Figure 5.2 is showing the x train of the given dataset. The given dataset is divided into the70-80%% part into the train dataset.

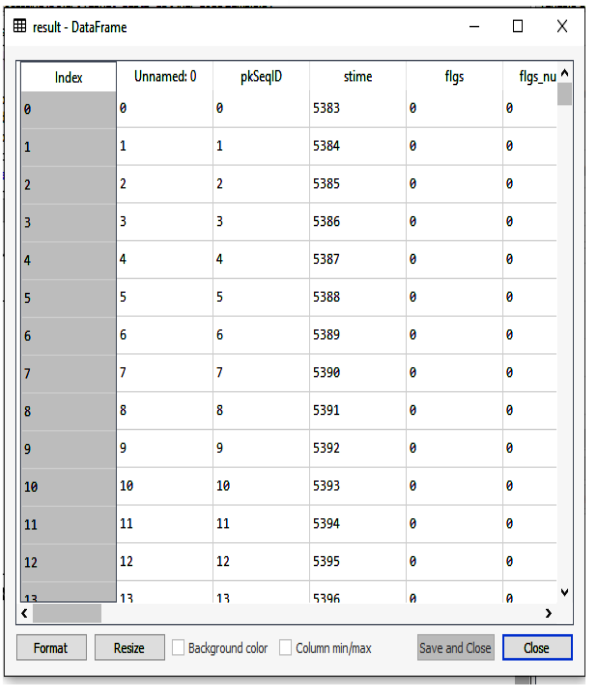

Figure 5.3: Result

Figure 5.3 presenting the result of the data frame after the preprocessing. The data is converting for the feature selection and the classification method.

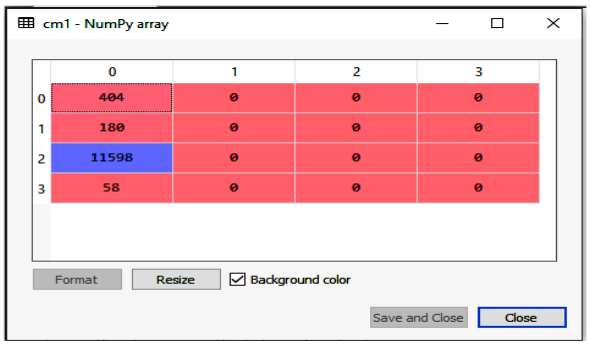

Figure 5.4: Confusion matrix LSTM

Figure 5.4 is presenting the confusion matrix of LSTM for the given dataset after the training and the testing.

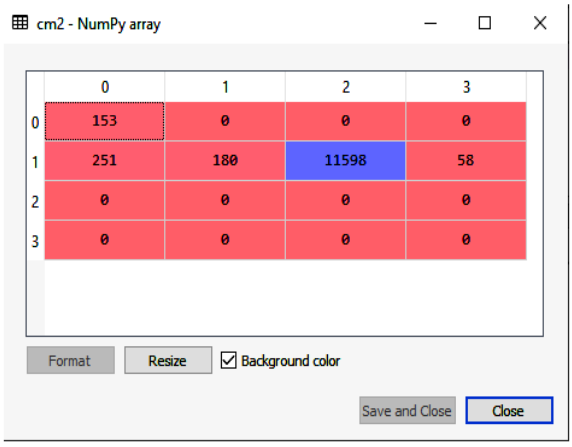

Figure 5.5: Confusion matrix of CNN

Figure 5.5 is presenting the confusion matrix of CNN for the given dataset after the trainingand the testing. It is matrix to identify the prediction of the given dataset.

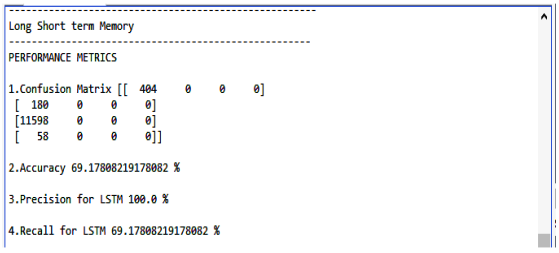

Figure 5.6: Performance of LSTM

Figure 5.6 is presenting long short term memory classification technique performance in term of the accuracy, precision and recall. The overall accuracy is 69.17%.

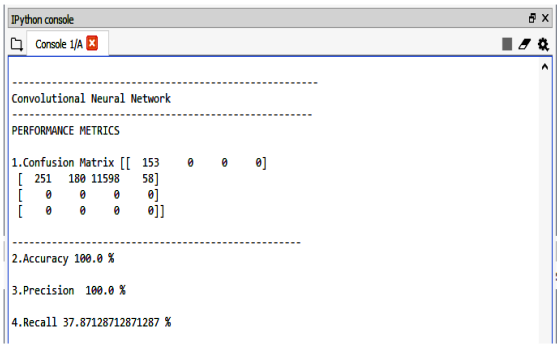

Figure 5.7: Performance of CNN

Figure 5.7 is presenting convolution neural network classification technique performance in term of the accuracy, precision and recall. The overall accuracy is 100%.

Sr. No.	Parameter Name	Value
1	Precision	100 %
2	Recall	69.17%
3	F_Measure	80 %
4	Accuracy	69.17%
5	Error Rate	30.83%

Table 5.1: Simulation Results of LSTM

Table 5.1 is showing the simulation results of the long short term memory technique. The overall accuracy is 69.17% with 30.83% error rate.

Sr. No.	Parameter Name	Value
1	Precision	100 %
2	Recall	37.87 %
3	F_Measure	90 %
4	Accuracy	100 %
5	Error Rate	Nil

Table 5.2: Simulation Results of CNN

Table 5.2 is showing the simulation results of the convolution neural network technique. The overall accuracy is 100% with 0% error rate.

Sr. No.	Parameters	Previous Work [1]	Proposed Work
1	Method	BLSTM	CNN
2	Precision	97 %	100 %
3	F_Measure	96 %	99 %
4	Accuracy	99.49 %	100 %
5	Error Rate	0.51%	Nil

Table 5.3: Result Comparison

Table 5.3 is showing the result comparison of the previous and proposed work. The precision of the proposed work is 97 % while in the previous work it is 100 %. Similarly the other parameter F_Measure is 99 % by the proposed work and 96 % by the previous work. The overall accuracy achieved by the proposed work is 100 % while previous it is achieved 99.49 %. The error rate of proposed technique is 0% while 0.51 % in existing works.

Therefore it is clear from the simulation results; the proposed work is achieved significant better results than existing work.

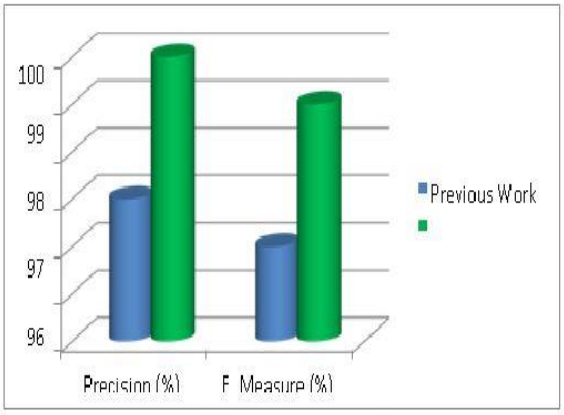

Figure 5.8: Result graph-parameter

Figure 5.8 is presenting the simulation results

values in the graphical form. The precision, and f measure is shown of the proposed a previous work.

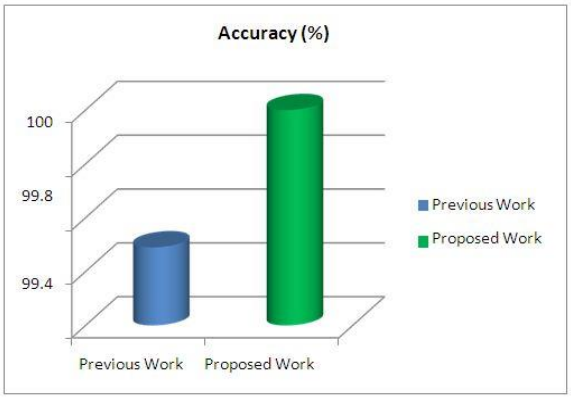

Figure 5.9: Accuracy

Figure 5.9 is presenting the simulation results graph of the accuracy. The proposed work achieved better accuracy then existing work.

4. CONCLUSION

A botnet attack is a massive scale digital attack carried out by malware-infected devices that are remotely controlled. It turns infected devices into zombie bots for the botnet regulator. Spam, data theft, the compromise of private information, sustained ad extortion, and more dangerous Distributed Denial of Service or DDoS attacks can all be carried out using botnet attacks [13]. Botnets are created by infecting several frameworks with malicious software and then handing them off to the botnet administrator to be used as slave frameworks. The PC framework can become familiar with this infection in a variety of ways, including: an email connection with a Trojan. An integrated system with direct communication between the Bot herder and each PC and a decentralised system with many links between all the infected computers.

This study made a case for adopting deep learning algorithms like LSTM and CNN for effective botnet detection in IoT networks [12]. By conducting extensive experiments with the most pertinent publically accessible dataset (Bot-IoT) in binary and multi-class classification scenarios, the efficacy of this strategy was confirmed. The Bot IoT was utilised as the dataset in this context due to its frequent updates, extensive attack variety, and different network protocols. We use the Bot-IoT dataset to assess our suggested strategy. Analysis of the simulation results revealed that, in comparison to the current BLSTM, our proposed method is effective and can, on average, produce better performance results.

Python Spyder 3.7 is used to run the simulation. The simulation results make it evident that the proposed work's precision is 97% as opposed to the old work's 100%. The other metric, F Measure, is also similar, being 99 percent by the suggested work and 96 percent by the earlier study. The suggested work achieves an overall accuracy of 100% as opposed to the prior study's 99.49%. The proposed technique has a 0% error rate compared to 0.51 l% in earlier efforts. Thus, it is evident from the simulation results that the suggested work has produced noticeably better outcomes than previous studies.

REFERENCES

1. S. I. Popoola, B. Adebisi, M. Hammoudeh, G. Gui land H. Gacanin, "Hybrid Deep Learning for Botnet Attack Detection in the Internet-of-Things Networks," in IEEE Internet of Things Journal, vol. 8, no. l6, pp. 4944-4956, l15 lMarch15, l2021, doi: 10.1109/JIOT.2020.3034156.

2. S. I. Popoola, R. Ande, B. Adebisi, G. Gui, M. Hammoudeh land O. Jogunola, "Federated Deep Learning for Zero-Day Botnet Attack Detection in IoT Edge Devices," in IEEE Internet of Things Journal, doi: 10.1109/JIOT.2021.3100755.

3. B. H. Schwengber, A. Vergütz, N. G. Prates land M. Nogueira, "Learning from Network Data Changes for Unsupervised Botnet Detection," in IEEE Transactions on Network land Service Management, doi: 10.1109/TNSM.2021.3109076.

4. F. Hussain et al., "A Two-Fold Machine Learning Approach to Prevent land Detect IoT

Botnet Attacks," in IEEE Access, vol. 9, lpp. 163412-163430, 2021, doi: 10.1109/ACCESS.2021.3131014.

5. R. Li, Q. Li, J. Zhou land Y. Jiang, "ADRIoT: An Edge-assisted Anomaly Detection Framework against IoT-based Network Attacks," in IEEE Internet of Things Journal, doi: 10.1109/JIOT.2021.3122148.

6. A. Alharbi land K. Alsubhi, "Botnet Detection Approach Using Graph-Based Machine Learning," in IEEE Access, vol. l9, pp. 99166-99180, 2021, doi: 10.1109/ACCESS.2021.3094183.

7. S. Qureshi et al., "A Hybrid DL-Based Detection Mechanism for Cyber Threats in Secure Networks," in IEEE Access, vol. 9, pp. 73938-73947, 2021, doi: 10.1109/ACCESS.2021.3081069.

8. W. N. H. Ibrahim let al., "Multilayer Framework for Botnet Detection Using Machine Learning Algorithms," in IEEE Access, vol. 9, pp. 48753-48768, 2021, doi: 10.1109/ACCESS.2021.3060778.

9. T. l-L. Wan let al., "Efficient Detection land Classification of Internet-of-Things Malware Based on Byte Sequences from Executable Files," in IEEE Open Journal of the Computer Society, vol. 1, pp. 262-275, 2020, doi: 10.1109/OJCS.2020.3033974.

10. L. Vu, V. L. Cao, Q. U. Nguyen, D. N. Nguyen, D. T. Hoang and E. Dutkiewicz, "Learning Latent Representation for IoT Anomaly Detection," in IEEE Transactions on Cybernetics, doi: 10.1109/TCYB.2020.3013416.

11. S. M. Sajjad, M. Yousaf, H. Afzal and M. R. Mufti, "eMUD: Enhanced Manufacturer Usage Description for IoT Botnets Prevention on Home WiFi Routers," in IEEE Access, vol. 8, pp. 164200-164213, 2020, doi: 10.1109/ACCESS.2020.3022272.

12. A. Blaise, M. Bouet, V. Conan and S. Secci, "Botnet Fingerprinting: A Frequency Distributions Scheme for Lightweight Bot Detection," in IEEE Transactions on Network and Service Management, vol. 17, no. 3, pp. 1701-1714, Sept. 2020, doi: 10.1109/TNSM.2020.2996502.

13. Y. Jia, F. Zhong, A. Alrawais, B. Gong and X. Cheng, "FlowGuard: An Intelligent Edge Defense Mechanism Against IoT DDoS Attacks," in IEEE internet of Things Journal, vol. 7, no. 10, pp. 9552-9562, Oct. 2020, doi:10.1109/JIOT.2020.2993782.

14. L. Silva, L. Utimura, K. Costa, M. Silva and S. Prado, "Study on Machine Learning Techniques for Botnet Detection," in IEEE Latin America Transactions, vol. 18, no. 05, pp. 881-888, May 2020, doi: 10.1109/TLA.2020.9082916.