International Journal of Research and Technology (IJRT)

International Open-Access, Peer-Reviewed, Refereed, Online Journal

ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529

| An ISO 9001:2015 Certified Journal |

Optimization Analysis of CNN-based Deep Learning System for Autonomous Detection of IoT Botnet Attacks

Raja Ram[1], Prof. Sanjay Pal[2]

M. Tech. Scholar, Department of Computer Science and Engineering, Oriental Institute of Science & Technology, Bhopal[1]

Assistant Professor, Department of Computer Science and Engineering, Oriental Institute of Science & Technology, Bhopal[2]

Abstract

The rapid growth of Internet of Things (IoT) devices has significantly increased the attack surface of modern networks, making them highly vulnerable to large-scale botnet attacks such as Distributed Denial of Service (DDoS), data exfiltration, and remote device takeover. Conventional intrusion detection systems based on signatures and handcrafted features are inadequate for identifying sophisticated and evolving IoT botnet behaviors in real time. To address these challenges, this paper presents an optimized Convolutional Neural Network (CNN)-based deep learning system for autonomous detection of IoT botnet attacks. This paper proposed for efficient botnet detection in IoT networks using deep learning algorithms such as LSTM and CNN. The effectiveness of this method was validated by performing extensive experiments with the most relevant publicly available dataset (Bot-IoT) in binary and multi-class classification scenarios. Simulation is performed using python spyder 3.7 software. It is clear from the simulation results the precision of the proposed work is 97 % whilein the previous work it is 100 %. Similarly, the other parameter F_Measure is 99 % by the proposed work and 96 % by the previous work.

Keywords: CNN, Botnet Attacks, Precision, F_measure

1. INTRODUCTION

The Internet of Things (IoT) has emerged as a key technological paradigm enabling seamless connectivity among billions of smart devices across applications such as smart homes, healthcare, industrial automation, transportation, and smart cities. While IoT systems improve efficiency and automation, their rapid growth has also introduced serious security vulnerabilities. Most IoT devices are resource-constrained, deployed with weak authentication mechanisms, limited encryption, and infrequent firmware updates, making them highly susceptible to cyberattacks. Among these threats, IoT botnet attacks have become one of the most damaging and widespread security challenges in modern networks. IoT botnets such as Mirai, Bashlite, and Mozi exploit compromised devices to create large-scale malicious networks capable of launching Distributed Denial of Service (DDoS) attacks, spreading malware, and disrupting critical services. Traditional intrusion detection systems (IDS), which rely on signature-based rules or handcrafted features, are often ineffective against these attacks due to their inability to adapt to evolving botnet behaviors and zero-day threats. Moreover, the dynamic and heterogeneous nature of IoT traffic further complicates accurate detection using conventional security mechanisms.

To overcome these limitations, machine learning (ML) and deep learning (DL) techniques have gained significant attention for IoT security. In particular, Convolutional Neural Networks (CNNs) have shown strong potential due to their ability to automatically extract hierarchical and spatial features from raw data. When applied to network traffic analysis, CNNs can learn complex patterns that distinguish normal IoT behavior from malicious botnet activities without requiring extensive manual feature engineering. However, deploying CNN-based detection systems in real-world IoT environments presents challenges related to model complexity, training efficiency, detection latency, and resource consumption [1, 2].

This research focuses on the optimization analysis of a CNN-based deep learning system for autonomous detection of IoT botnet attacks [3, 4]. The proposed approach investigates optimization strategies at multiple levels, including feature preprocessing, hyperparameter tuning, architectural refinement, and training regularization, to achieve high detection accuracy with reduced computational overhead. By designing a lightweight yet effective CNN model, the system enables real-time and autonomous monitoring of IoT network traffic, making it suitable for edge and fog computing scenarios [5].

The primary objective of this work is to develop a scalable and intelligent intrusion detection framework that can accurately identify IoT botnet attacks while adapting to evolving threat patterns. The contributions of this study aim to enhance IoT cybersecurity by providing an optimized, autonomous, and robust deep learning-based solution for protecting large-scale IoT infrastructures [6, 7].

2. CNN

A Convolutional Neural Network (CNN) is a class of deep learning models specifically designed to automatically extract meaningful features from structured data such as images, signals, and time-series representations. In the context of IoT botnet attack detection, CNNs are widely adopted due to their strong capability to learn complex and non-linear patterns from network traffic data without relying on manual feature engineering.

CNN Architecture

A typical CNN architecture consists of several key layers:

1. Input Layer: The input layer receives preprocessed IoT network traffic data. Traffic features such as packet size, flow duration, protocol type, source and destination ports are transformed into 2D feature maps or matrices suitable for convolution operations.
2. Convolutional Layer: The convolutional layer applies multiple learnable filters (kernels) to the input feature maps. These filters slide across the input to capture local spatial correlations and hidden patterns associated with normal and malicious IoT traffic behavior. In botnet detection, convolution layers are effective in identifying attack signatures embedded within traffic flows.
3. Activation Function: Non-linear activation functions, commonly Rectified Linear Unit (ReLU), are applied to introduce non-linearity into the model. This enables the CNN to learn complex relationships between features and improves detection accuracy.
4. Pooling Layer: Pooling layers, such as max pooling or average pooling, reduce the spatial dimensions of feature maps while preserving important information. This

reduces computational complexity and helps prevent overfitting, which is critical for real-time IoT environments.

5. Fully Connected Layer: The fully connected layers combine the extracted features from convolutional layers and perform high-level reasoning. These layers classify the input traffic as either normal or botnet-infected.
6. Output Layer: The output layer uses activation functions such as Softmax or Sigmoid to produce the final classification results, indicating the probability of an IoT botnet attack.

Role of CNN in IoT Botnet Detection

CNNs enable autonomous detection by learning attack patterns directly from data rather than relying on predefined rules. Their ability to detect spatial dependencies in traffic features allows them to identify both known and previously unseen botnet attacks. Moreover, CNNs can be optimized to create lightweight architectures suitable for deployment at the network edge, ensuring low detection latency and efficient resource utilization.

3. PROPOSED METHODOLOGY

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

- To collect bot-IOT dataset from kaggle machine learning resoprocity.

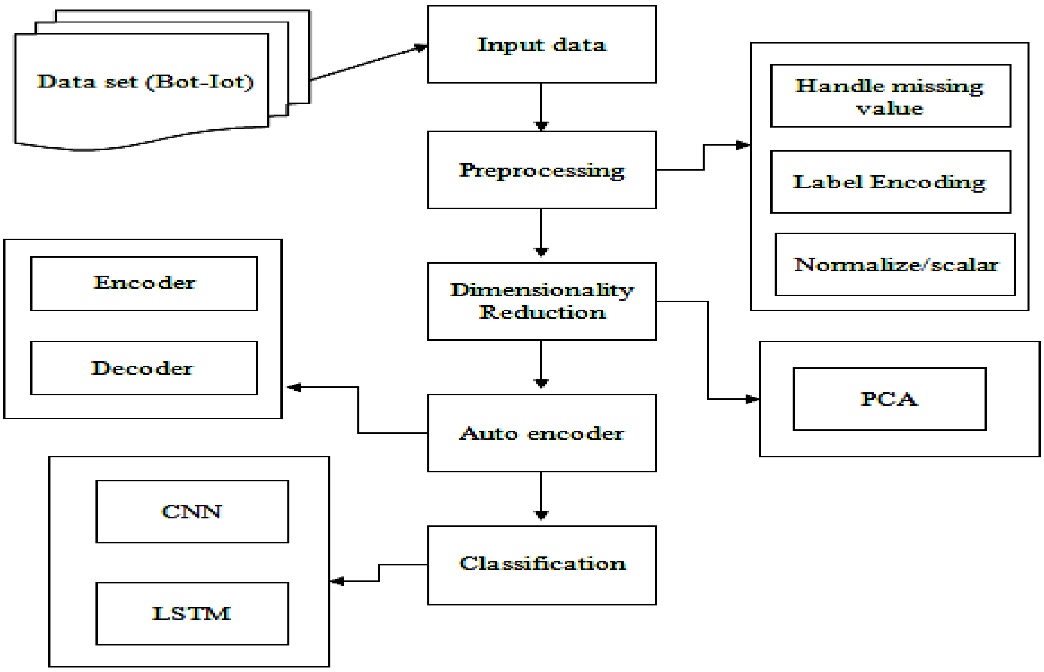


Figure 1: Flow Chart

- To implement the deep learning algorithm such as LSTM and CNN, for better performance.
- To implement the feature dimensionality reduction such as Principle Component Analysis (PCA), for reducing the number of dimensions in the dataset.
- To implement the auto encoder for compressing the raw data's.
- To check the performance parameters and enhance.

Step-

1. In this system, the Bot-Iot dataset will be taken as input from the dataset repository.
2. Then, we have to implement the data preprocessing step. In this step, we have to handle the missing values for avoid wrong prediction, to encode the label for input data and normalize/ scaling the input data.
3. Then, we have to implement the feature dimensionality reduction such as Principal Component Analysis (PCA).
4. We have to implement the deep learning algorithms such as Long Short term Memory (LSTM) and Convolutional Neural Network (CNN).
5. Finally, the simulation results shows that the performance metrics such as accuracy, precision, recall and confusion matrix value will be improved

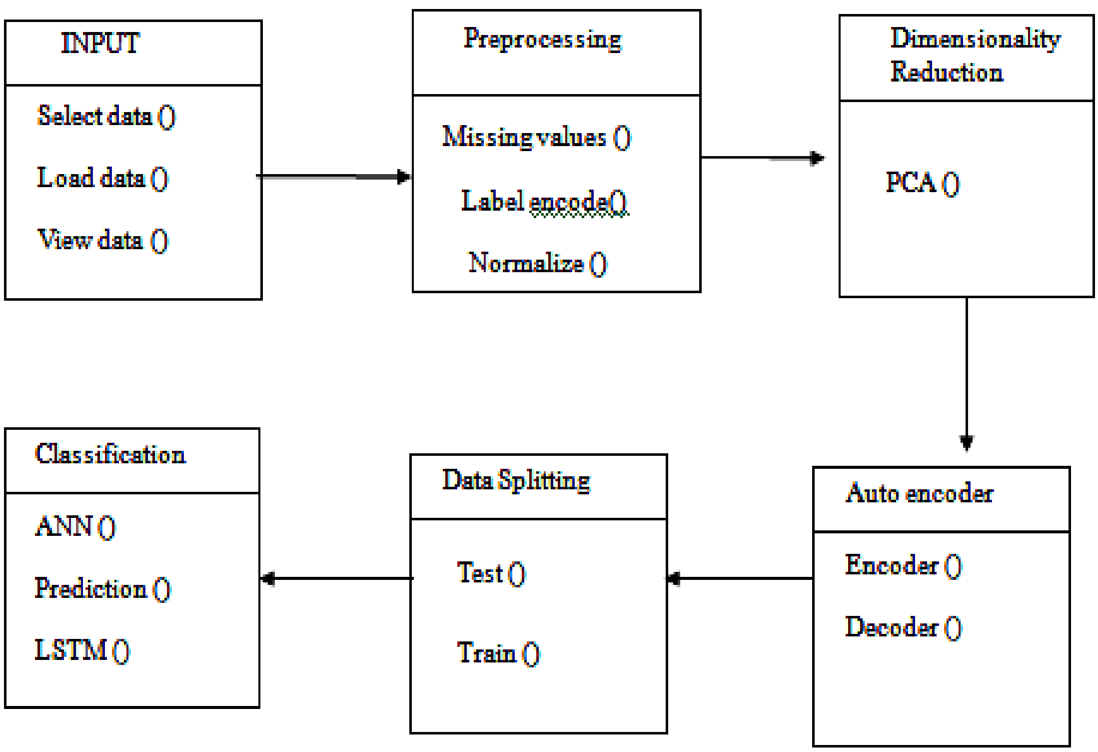


Figure 2: Class Diagram

Figure 2 is presenting the class diagram of the proposed model. The various step in thismodel make complete the prediction work.

International Journal of Research and Technology (IJRT)

International Open-Access, Peer-Reviewed, Refereed, Online Journal

ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529

| An ISO 9001:2015 Certified Journal |

4. SIMULATION RESULTS

The simulation starts from taking the dataset. In this dataset the various features valuemention like pkSeqID stime, flgs, flgs_number, proto, proto_number, saddr, sport daddr, dport pkts, bytes, statestate_number ltime, seq, duretc.

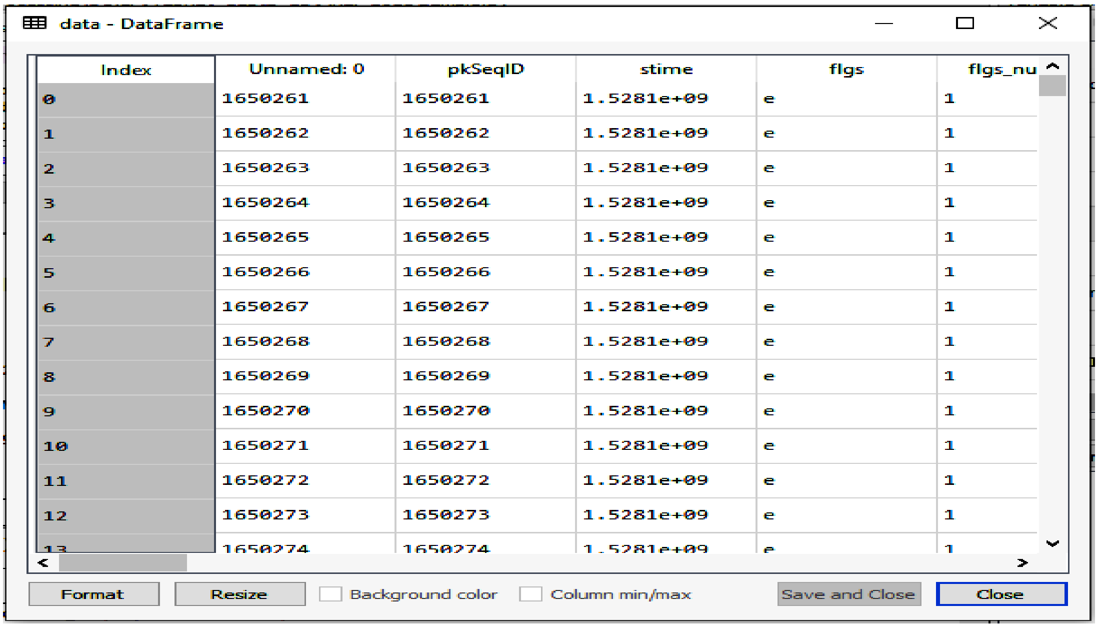
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC
1	No	pkSeqI	stime	flgs	nu	rot	n	saddr	sport	daddr	dport	pkts	bytes	state	te_num	ltime	seq	dur	mean	stddev	sum	min	max	spkts	dpkts	sbytes	dbytes	rate	srate
2	2E+06	2E+06	2E+09	e	1	tcp	1	192.168.100.1	54110	192.168	80	10	1729	RST	1	2E+09	20	6.4064	0.6795	0.5441	1.3589	0.1353	1.2236	6	4	963	766	1.4048	0.7805
3	2E+06	2E+06	2E+09	e	1	tcp	1	192.168.100.1	54112	192.168	80	10	1604	RST	1	2E+09	21	6.4059	0.6796	0.5442	1.3591	0.1354	1.2238	6	4	838	766	1.405	0.7805
4	2E+06	2E+06	2E+09	e	1	tcp	1	192.168.100.1	54114	192.168	80	8	1708	RST	1	2E+09	22	6.401	1.1108	1.1108	2.2217	0	2.2217	5	3	1008	700	1.0936	0.6249
5	2E+06	2E+06	2E+09	e	1	tcp	1	192.168.100.1	54116	192.168	80	8	1462	RST	1	2E+09	23	6.4007	1.1133	1.1133	2.2267	0	2.2267	5	3	762	700	1.0936	0.6249
6	2E+06	2E+06	2E+09	e	1	tcp	1	192.168.100.1	54118	192.168	80	8	1296	RST	1	2E+09	24	6.4005	1.1131	1.1131	2.2262	0	2.2262	5	3	596	700	1.0937	0.625
7	2E+06	2E+06	2E+09	e	1	tcp	1	192.168.100.1	54120	192.168	80	8	1434	RST	1	2E+09	25	6.4002	1.1134	1.1134	2.2268	0	2.2268	5	3	734	700	1.0937	0.625
8	2E+06	2E+06	2E+09	e	1	tcp	1	192.168.100.1	54122	192.168	80	8	1764	RST	1	2E+09	26	6.3996	1.6129	1.6129	3.2259	0	3.2259	5	3	1064	700	1.0938	0.625
9	2E+06	2E+06	2E+09	e	1	tcp	1	192.168.100.1	54124	192.168	80	8	1469	RST	1	2E+09	27	6.3994	1.6132	1.6132	3.2263	0	3.2263	5	3	769	700	1.0938	0.6251
10	2E+06	2E+06	2E+09	e	1	tcp	1	192.168.100.1	54126	192.168	80	8	1613	RST	1	2E+09	28	6.3964	1.612	1.612	3.2239	0	3.2239	5	3	913	700	1.0944	0.6254
11	2E+06	2E+06	2E+09	e	1	tcp	1	192.168.100.1	54128	192.168	80	8	1376	RST	1	2E+09	29	6.3962	1.6128	1.6128	3.2256	0	3.2256	5	3	676	700	1.0944	0.6254
12	2E+06	2E+06	2E+09	e	1	tcp	1	192.168.100.1	54130	192.168	80	8	1458	RST	1	2E+09	30	6.3959	1.613	1.613	3.2261	0	3.2261	5	3	758	700	1.0945	0.6254
13	2E+06	2E+06	2E+09	e	1	tcp	1	192.168.100.1	54132	192.168	80	8	1391	RST	1	2E+09	31	6.3956	1.6146	1.6146	3.2291	0	3.2291	5	3	691	700	1.0945	0.6254
14	2E+06	2E+06	2E+09	e	1	tcp	1	192.168.100.1	54134	192.168	80	8	1408	RST	1	2E+09	32	6.3953	1.6148	1.6148	3.2296	0	3.2296	5	3	708	700	1.0946	0.6255
15	2E+06	2E+06	2E+09	e	1	tcp	1	192.168.100.1	54136	192.168	80	8	1406	RST	1	2E+09	33	6.3951	1.6151	1.6151	3.2302	0	3.2302	5	3	706	700	1.0946	0.6255
16	2E+06	2E+06	2E+09	e	1	tcp	1	192.168.100.1	54138	192.168	80	8	1415	RST	1	2E+09	34	6.3948	2.1163	2.1163	4.2327	0	4.2327	5	3	715	700	1.0946	0.6255
17	2E+06	2E+06	2E+09	e	1	tcp	1	192.168.100.1	54140	192.168	80	8	1517	RST	1	2E+09	35	6.3945	2.1165	2.1165	4.2331	0	4.2331	5	3	817	700	1.0947	0.6255
18	2E+06	2E+06	2E+09	e	1	tcp	1	192.168.100.1	54142	192.168	80	8	1363	RST	1	2E+09	36	6.3928	2.116	2.116	4.2319	0	4.2319	5	3	663	700	1.095	0.6257
19	2E+06	2E+06	2E+09	e	1	tcp	1	192.168.100.1	54144	192.168	80	8	1643	RST	1	2E+09	37	6.3924	2.1161	2.1161	4.2323	0	4.2323	5	3	943	700	1.0951	0.6257
20	2E+06	2E+06	2E+09	e	1	tcp	1	192.168.100.1	54146	192.168	80	8	1570	RST	1	2E+09	38	6.3902	2.1198	2.1198	4.2396	0	4.2396	5	3	870	700	1.0954	0.626
21	2E+06	2E+06	2E+09	e	1	tcp	1	192.168.100.1	54148	192.168	80	8	1657	RST	1	2E+09	39	6.3898	2.1156	2.1156	4.2311	0	4.2311	5	3	957	700	1.0955	0.626
22	2E+06	2E+06	2E+09	e	1	tcp	1	192.168.100.1	54150	192.168	80	8	1590	RST	1	2E+09	40	6.3895	2.1157	2.1157	4.2315	0	4.2315	5	3	890	700	1.0956	0.626
23	2E+06	2E+06	2E+09	e	1	tcp	1	192.168.100.1	54152	192.168	80	8	1493	RST	1	2E+09	41	6.3892	2.116	2.116	4.2319	0	4.2319	5	3	793	700	1.0956	0.6261
24	2E+06	2E+06	2E+09	e	1	tcp	1	192.168.100.1	54154	192.168	80	8	1615	RST	1	2E+09	42	6.3864	2.115	2.115	4.2299	0	4.2299	5	3	915	700	1.0961	0.6263
25	2E+06	2E+06	2E+09	e	1	tcp	1	192.168.100.1	54156	192.168	80	8	1486	RST	1	2E+09	43	6.3862	2.1177	2.1177	4.2354	0	4.2354	5	3	786	700	1.0961	0.6264

DDoSdata

Figure 3: Original dataset in .csv file

The figure 3 is showing the dataset, which is taken from the kaggle machine learning website. Figure 4 is showing the dataset in the python environment. The dataset have variousnumbers of rows and column. The features name is mention in each column. Table 1 is showing the simulation results of the convolution neural network technique. Theoverall accuracy is 100% with 0% error rate.

International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |



Index	Unnamed: 0	pkSeqID	stime	flgs	flgs_nu
0	1650261	1650261	1.5281e+09	e	1
1	1650262	1650262	1.5281e+09	e	1
2	1650263	1650263	1.5281e+09	e	1
3	1650264	1650264	1.5281e+09	e	1
4	1650265	1650265	1.5281e+09	e	1
5	1650266	1650266	1.5281e+09	e	1
6	1650267	1650267	1.5281e+09	e	1
7	1650268	1650268	1.5281e+09	e	1
8	1650269	1650269	1.5281e+09	e	1
9	1650270	1650270	1.5281e+09	e	1
10	1650271	1650271	1.5281e+09	e	1
11	1650272	1650272	1.5281e+09	e	1
12	1650273	1650273	1.5281e+09	e	1
13	1650274	1650274	1.5281e+09	e	1

Figure 4: Dataset

Table 1: Result Comparison

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

Figure 5 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.

International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

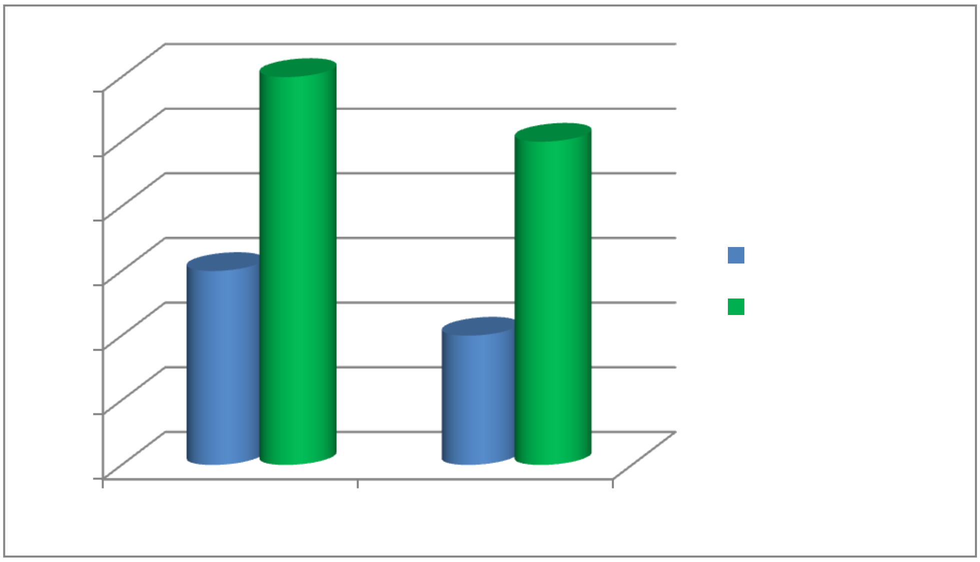


Figure 5: Result graph-parameters

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

5. CONCLUSIONS

This study presented an optimized CNN-based deep learning system for the autonomous detection of IoT botnet attacks, addressing the critical security challenges posed by the rapid proliferation of IoT devices. By leveraging the automatic feature extraction capability of Convolutional Neural Networks, the proposed system effectively learns complex traffic patterns associated with malicious botnet behavior, overcoming the limitations of traditional signature-based and classical machine learning intrusion detection approaches.

The optimization analysis demonstrated that careful tuning of hyperparameters, refinement of network architecture, and the application of regularization techniques such as dropout, batch normalization, and early stopping significantly enhance detection accuracy while reducing false positives and computational overhead. Additionally, feature normalization and lightweight CNN design contributed to faster convergence and improved real-time performance, making the system suitable for deployment in resource-constrained IoT, edge, and fog computing environments.

REFERENCES

1. S. I. Popoola, B. Adebisi, M. Hammoudeh, G. Gui and H. Gacanin, "Hybrid Deep Learning for Botnet Attack Detection in the Internet-of-Things Networks," in IEEE Internet of Things Journal, vol. 8, no. 6, pp. 4944-4956, 15 March15, 2021, doi: 10.1109/JIOT.2020.3034156.

2. 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, doi: 10.1109/JIOT.2021.3100755.
3. B. H. Schwengber, A. Vergütz, N. G. Prates and M. Nogueira, "Learning from Network Data Changes for Unsupervised Botnet Detection," in IEEE Transactions on Network and Service Management, doi: 10.1109/TNSM.2021.3109076.
4. F. Hussain et al., "A Two-Fold Machine Learning Approach to Prevent and Detect IoT Botnet Attacks," in IEEE Access, vol. 9, pp. 163412-163430, 2021, doi: 10.1109/ACCESS.2021.3131014.
5. R. Li, Q. Li, J. Zhou and 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. B. H. Schwengber, A. Vergütz, N. G. Prates and M. Nogueira, "Learning from Network Data Changes for Unsupervised Botnet Detection," in IEEE Transactions on Network and Service Management, doi: 10.1109/TNSM.2021.3109076.
7. A. Alharbi and K. Alsubhi, "Botnet Detection Approach Using Graph-Based Machine Learning," in IEEE Access, vol. 9, pp. 99166-99180, 2021, doi: 10.1109/ACCESS.2021.3094183.
8. 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.
9. W. N. H. Ibrahim et 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.
10. T. -L. Wan et al., "Efficient Detection and 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.
11. 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.
12. 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.
13. 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.
14. 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.
15. 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.