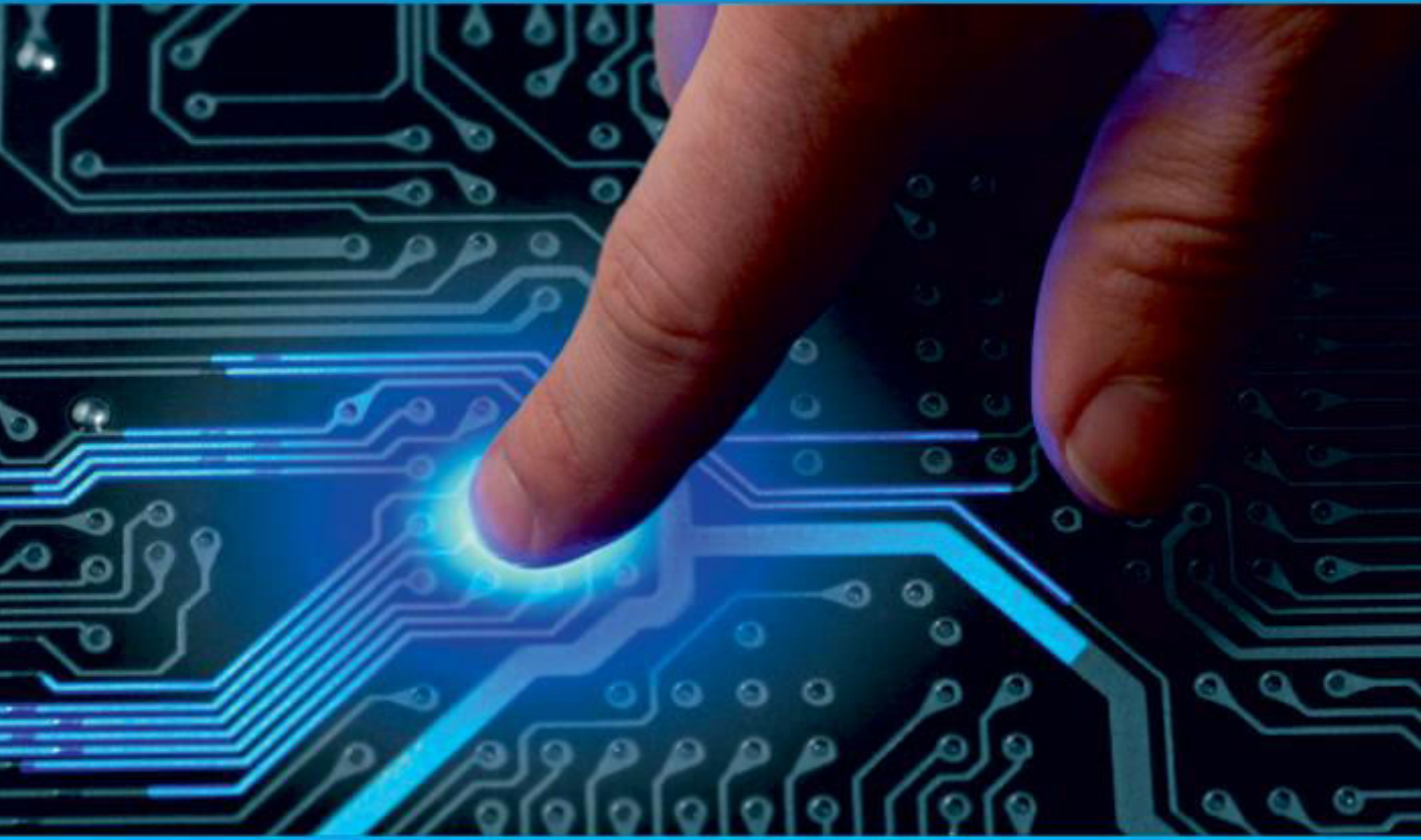
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 4, April 2024

INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.379

| DOI: 10.15680/IJIRCCE.2024.1204134 |

Securing the IoT Ecosystem: A Multitask Deep Learning Framework for Malware Detection via Behavioral Traffic Analysis

Dr. B. V. V. Siva Prasad, A. Rajavamshi Goud, K. Vinay, T. Ajay

Associate Professor, Department of Computer Science and Engineering, Anurag University, Hyderabad, Telangana, India
B.Tech Student, Department of Computer Science and Engineering, Anurag University, Hyderabad, Telangana, India
B.Tech Student, Department of Computer Science and Engineering, Anurag University, Hyderabad, Telangana, India
B.Tech Student, Department of Computer Science and Engineering, Anurag University, Hyderabad, Telangana, India

ABSTRACT: The proliferation of Internet of Things (IoT) technology brings numerous benefits but also introduces new security challenges, particularly with the rise of IoT-specific malware orchestrating large-scale attacks through compromised devices. Safeguarding IoT systems and mitigating the expansion of malware threats are imperative. To evaluate IoT system security and enhance cyber defense mechanisms, comprehensive data collection and analysis from diverse IoT sources are essential. In response, this research proposes a multitask Deep Learning (DL) model tailored for IoT malware detection. Leveraging Long Short-Term Memory (LSTM), our model adeptly addresses two critical tasks: discerning the nature of incoming traffic as benign or malicious, and identifying the specific type of malware present within malicious network activity. The model's efficacy is demonstrated through extensive experimentation using a substantial dataset comprising 145 .pcap files capturing both benign and malicious traffic from 18 distinct IoT devices. Time-series analysis of traffic flows informs the model's training process. Features extracted from the dataset are classified into three modalities: flow-related, traffic flag-related, and packet payload-related. Employing a rigorous feature selection strategy at both feature and modality levels, we identify the most discriminative features for enhancing model performance. Notably, our findings indicate that flow-related features excel in task 1, while flag-related features yield superior accuracy in task 2 and multitask classification. This research underscores the potential of multitask DL models in fortifying IoT security against evolving malware threats, offering insights into effective strategies for cyber defense in IoT environments..

KEYWORDS: Internet of Things, Malware Detection, Natural language, Long Short-Term Memory

I. INTRODUCTION

Introduction: The Internet of Things (IoT) has revolutionized various domains, including automobiles, smart homes and cities, manufacturing, healthcare, retail, and beyond. With the exponential growth of portable Internet-connected devices, the technical sophistication and manufacturing simplicity of these intelligent devices continue to evolve. However, this advancement also introduces a myriad of security challenges. The 2020 IoT threat report underscores the pressing need for heightened security measures due to the emergence of IoT-specific malware, exemplified by the notorious Mirai botnet.

The Mirai botnet gained infamy for orchestrating one of the largest Distributed Denial-of-Service (DDoS) attacks on record, targeting Dyn. Moreover, the availability of the Mirai source code has catalyzed the development of sophisticated malware variants like Satori, Hajime, and BrickerBot. Consequently, researchers have intensified efforts to develop robust solutions to combat IoT security threats. However, the lack of empirical data on current IoT malware and a comprehensive understanding of malware-infected device behaviors pose significant challenges.

This study identifies two distinct challenges in IoT security: protecting devices against malware attacks and classifying IoT malware based on generated traffic. While the primary objective remains device protection, the evolution of highly sophisticated ransomware renders complete device security unattainable. Hence, it becomes imprecate to classify different malware types, enabling targeted mitigation strategies.

Existing literature emphasizes the relevance of IoT vulnerabilities and underscores the need for enhanced intrusion detection algorithms. However, validating the efficacy of these algorithms remains challenging due to the unique characteristics of IoT traffic. Traditional Machine Learning (ML) approaches, although effective, require domain expertise and extensive feature engineering, limiting their scalability. In contrast, Deep Learning (DL) techniques offer the advantage of learning features directly from raw data, bypassing the need for manual feature extraction.

To address these challenges, this research proposes a Long Short-Term Memory (LSTM) DL-based multitask intrusion detection model. By leveraging heterogeneous dataset characteristics, including various IoT devices and attack types, the proposed model aims to enhance intrusion detection accuracy and robustness. This study contributes by proposing a multitask LSTM-based DL model for IoT security and malware classification, investigating the functionality of LSTM models with heterogeneous time series data, and evaluating the proposed model's performance through comprehensive experiments. checked at all times from everyone This is because the information is publicly available and distributed globally. It is chronologically updated and cryptographically sealed. The full range of applicable use cases for this technology

II. LITERATURE REVIEW

The Internet of Things (IoT) has witnessed exponential growth, permeating various sectors such as automotive, healthcare, manufacturing, and smart cities. However, this proliferation of IoT devices has also ushered in a new era of cybersecurity challenges. Notably, the emergence of IoT-specific malware, exemplified by the Mirai botnet, has underscored the vulnerability of IoT ecosystems to large-scale cyberattacks . As such, research efforts have intensified to address these security concerns, focusing on two primary areas: IoT security against malware attacks and classification of IoT malware based on traffic patterns.

Datasets play a crucial role in evaluating intrusion detection technologies and developing effective malware classification techniques. However, existing datasets often fail to capture the diverse characteristics of modern IoT devices and the evolving nature of cyber threats. While datasets like NSL-KDD and CTU-13 are commonly used for general network traffic analysis, they may not adequately represent IoT-specific traffic. Several pioneering IoT datasets, such as IoT network intrusion and N-BaIoT, have attempted to bridge this gap but are still limited in scope. Machine learning (ML)-based approaches offer promise in distinguishing between normal and malicious traffic patterns. Supervised ML algorithms, including support vector machines (SVM) and ensemble methods, have demonstrated success in detecting IoT-related attacks. Unsupervised ML techniques, such as auto-encoders, provide avenues for anomaly detection in IoT environments.

Deep learning (DL) algorithms have emerged as powerful tools for malware traffic monitoring and classification in the context of IoT security. Several studies have explored different DL architectures for this purpose, each leveraging unique approaches to analyze and classify IoT-related traffic. Gao et al. employed deep belief networks to tackle malware traffic monitoring and classification. By utilizing deep belief networks, they aimed to capture complex patterns in IoT traffic data, enabling accurate detection of malicious activity. Similarly, Shone et al proposed a novel approach that integrated non-symmetric deep auto-encoders with a random forest, termed a sparse auto-encoder. This innovative architecture enabled effective extraction of features from IoT traffic data, enhancing the detection and classification of malware. Bendiab et al. utilized recurrent neural networks (RNNs) to learn the temporal characteristics of IoT traffic, converting them into a series of characters. This approach enabled the modeling of temporal dependencies in IoT traffic data, improving the accuracy of malware detection. In contrast, Shire et al. employed the convolutional neural network (CNN) MobileNet to analyze IoT malware traffic. By leveraging CNNs, they aimed to capture spatial features in IoT traffic data, enabling robust classification of malware types.

Besides DL-based approaches, researchers have explored alternative techniques for IoT malware detection and classification:
Entropy Measures: François et al. [48] utilized entropy measures to detect large-scale abnormalities in network traffic. By comparing typical traffic profiles with incoming flow data, they identified anomalies, offering a complementary approach to traditional detection methods. Markov Chains: García et al. [49] employed Markov Chains to simulate states within the command-and-control channel. This approach provides a behavioral botnet detection mechanism by modeling botnet communication dynamics, aiding in the identification of malicious activities within IoT networks.
DNS Activity Monitoring: Singh et al. [50] focused on detecting bot-infected devices by monitoring DNS activity at an enterprise level. Through regular analysis of DNS logs, they aimed to identify and isolate compromised devices, mitigating the impact of botnet infections within IoT environments.
String-Based Feature Analysis: Dib et al. [51] utilized string-based features for malware classification and clustering. By

extracting and analyzing string-based attributes from malware samples, they identified common characteristics and grouped similar malware types together, facilitating efficient threat analysis and response strategies.

III. EXISTING SYSTEM WITH DISADVANTAGES

In the literature, an approach was introduced for the identification and classification of network devices, particularly IoT devices, and their corresponding traffic. This approach employs a supervised learning algorithm known as Random Forest, which analyzes the contents of network packets to identify devices and classify traffic based on the generating application. The primary objective is to facilitate network analysis, management, access control, provisioning, and resource allocation.

Another research effort evaluated the robustness of classifiers within three input space categories against three proposed attacks. The experiments conducted aimed to analyze the performance of classifiers in the face of adversarial network traffic (ANT) and discuss practical implications. The focus of this research lies in utilizing network traffic, DL-based classifiers, and adversarial examples within DL-based network traffic classification and ANT. Additionally, it was demonstrated that generating universal adversarial perturbation does not necessarily require extensive data, and such perturbations are not unique.

Disadvantages of Existing System:

Dependency on Supervised Learning: The existing approach heavily relies on Random Forest, a supervised learning algorithm, for device identification and traffic classification. Although Random Forest is robust, it may struggle to accurately identify intricate patterns and variations in network traffic, particularly when compared to more sophisticated deep learning techniques like LSTM employed in our work.

Lack of Malware Detection Focus: While the existing approach addresses device identification and traffic classification, it overlooks the critical aspect of malware detection in IoT environments. This limitation could pose significant security risks in scenarios where IoT devices are vulnerable to malware attacks.

Inefficient Packet Inspection: The existing approach inspects network packet contents for device identification and traffic classification. However, this method may face challenges in efficiently handling large-scale data and extracting meaningful features, especially in complex IoT malware detection scenarios.

Scalability Concerns: Random Forest may exhibit limitations in scalability and resource usage, particularly when confronted with large volumes of network traffic data. This could hinder its practical applicability in real-world IoT environments with extensive data streams.

In summary, while the existing system provides valuable insights into network device identification and traffic classification, addressing the aforementioned limitations is crucial for enhancing its effectiveness, especially in the context of IoT security and malware detection.

IV. PROPOSED METHOD

We propose a multitask deep learning (DL) model for detecting IoT malware. Our approach involves not only Long Short-Term Memory (LSTM) but also Convolutional Neural Network (CNN) and a hybrid CNN+LSTM model. These models efficiently perform two tasks: determining whether the provided traffic is benign or malicious, and identifying the specific malware type present in the malicious network traffic. To train our models, we utilized large-scale traffic data comprising both benign and malicious traffic collected from 18 different IoT devices.

In our methodology, we conducted a comprehensive time-series analysis on the packets of traffic flows. These time-series data were then used to train the proposed DL models. The extracted features from the dataset were categorized into three modalities: flow-related, traffic flag-related, and packet payload-related features.

To enhance model performance, we employed a feature selection approach at both the feature and modality levels. This allowed us to identify and utilize the most informative features and modalities.

We conducted several experiments, including single-task and multitasking scenarios, to assess the performance of our proposed models. These experiments involved: Applying class imbalance techniques to handle the uneven distribution of

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 4, April 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1204134 |

benign and malicious samples. Employing feature selection methods to identify the most relevant features for classification. Fusing modalities of the time-series network traffic data to leverage complementary information.
Through extensive experimentation and evaluation, we compared the performance of LSTM, CNN, and CNN+LSTM models. Our objective was to choose the algorithm that provided the highest accuracy for malware detection in IoT network traffic.

The proposed framework for IoT malware detection involves several key components and stages:

1. Data Collection and Splitting into Modalities:
Raw IoT concatenated dataset is split into different modalities such as flow, flag, and packets. Each modality has a different number of features: flow (31), flag (21), and packets (28).

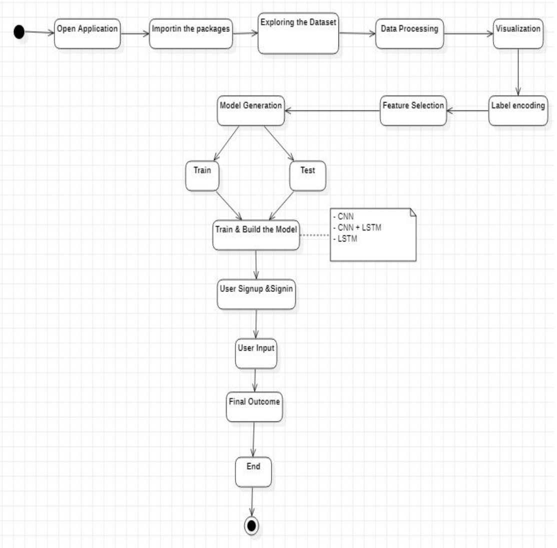


Fig 1: Sequence diagram of the proposed model

2. Data Preprocessing:
Missing values in all modalities are handled, with features having a missing data percentage exceeding 30% eliminated. Dataset is split into training set (80%) and testing set (20%).
Data is rescaled to [0,1] range using Min-Max normalization to make it uniform for model convergence. Outliers are identified and replaced with each class's average value. Data balancing is performed to address imbalanced datasets, with Synthetic Minority Oversampling Technique (SMOTE) and its variants used. SMOTE Edited Nearest Neighbor (SMOTEENN) method is selected to handle class imbalance in the training set, while testing set is balanced by oversampling.

3. Feature Selection:
Feature selection is crucial for model performance. Two approaches are tested: early feature selection on the concatenated IoT dataset and late feature selection on individual modalities. Late feature selection strategy is adopted to account for interdependencies among modalities. Recursive XGBoost and SULOV methods are employed to reduce features and select the best ones.

4. Model Training and Evaluation:
Keras framework with TensorFlow backend is used to implement the models. Binary and multi-class classification tasks are performed using ReLU activation function with binary and categorical cross-entropy loss, respectively. Adam optimizer with 0.001 learning rate optimization is utilized. Model comprises a single LSTM layer with 128 hidden units for feature extraction, followed by l2-norm and dropout of 0.01 each.Three dense layers with 64, 32, and 32 hidden units

are used for both tasks. Training parameters include 100 epochs and a batch size of 240. Model performance is evaluated using accuracy, precision, recall, and F1-score metrics.

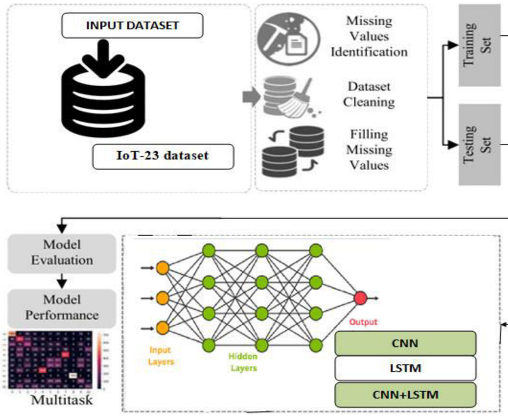


Fig 2: Model Architecture

Evaluation metrics:

The proposed multitask model's classification performance is evaluated using four key metrics:

1. Accuracy: This metric measures the percentage of instances correctly classified out of the total number of instances. It is calculated using the formula:

Accuracy = $(TP + TN)/(TP + TN + FP + FN)$

2. Precision: Precision measures the percentage of accurately classified positive instances out of all instances predicted as positive. It is calculated using the formula:

Precision= $TP/(TP + FP)$

3. Recall: Recall, also known as sensitivity or true positive rate, measures the percentage of accurately classified positive instances out of all actual positive instances. It is calculated using the formula:

Recall = $TP/(TP + FN)$

4. F1-Score: The F1-Score combines precision and recall into a single metric, providing a balance between the two. It is calculated as the harmonic mean of precision and recall:

F1-score = (2 × Precision × Recall)/(Precision + Recall)

Where:

T P (True Positive) represents the number of instances correctly classified as positive.
T N (True Negative) represents the number of instances correctly classified as negative.
FP (False Positive) represents the number of instances incorrectly classified as positive.
FN (False Negative) represents the number of instances incorrectly classified as negative.

These metrics provide comprehensive insights into the model's performance, considering both its ability to correctly classify instances (accuracy) and its ability to minimize false positives and false negatives (precision, recall, and F1-Score).

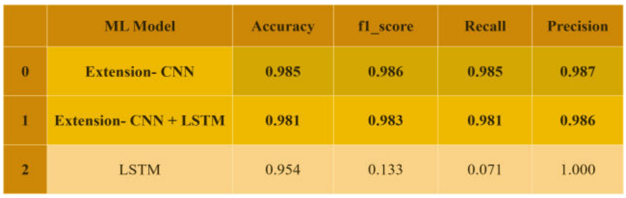
Algorithms:

CNN: Convolutional Neural Network (CNN) is a type of deep learning algorithm that is particularly well-suited for image recognition and processing tasks. It is made up of multiple layers, including convolutional layers, pooling layers, and fully connected layers.

CNN + LSTM: LSTM networks extend the recurrent neural network (RNNs) mainly designed to deal with situations in which RNNs do not work. When we talk about RNN, it is an algorithm that processes the current input by taking into account the output of previous events (feedback) and then storing it in the memory of its users for a brief amount of time (short-term memory). Of the many applications, its most well-known ones are those in the areas of non-Markovian speech control and music composition. However, there are some drawbacks to RNNs.

V. RESULT

In our implementation, we explored three different deep learning models: Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and a hybrid CNN+LSTM architecture. Initially, we trained the LSTM model and achieved an accuracy of approximately 90%. While this accuracy was acceptable, we sought to improve upon it by evaluating the performance of CNN and CNN+LSTM models.



	ML Model	Accuracy	f1_score	Recall	Precision
0	Extension- CNN	0.985	0.986	0.985	0.987
1	Extension- CNN + LSTM	0.981	0.983	0.981	0.986
2	LSTM	0.954	0.133	0.071	1.000

Fig 3: The comparison between CNN, LSTM and CNN+LSTM models

Upon experimentation, we found that both CNN and CNN+LSTM models consistently outperformed the LSTM model. Specifically, both CNN and CNN+LSTM models yielded accuracies nearing 98%, indicating a substantial improvement over the LSTM baseline. Among the three models, CNN emerged as the top performer, demonstrating its dominance in accurately classifying IoT network traffic as benign or malicious.

This evaluation process allowed us to select CNN as the preferred model for our IoT malware detection task due to its superior accuracy. By leveraging CNN's capabilities in extracting hierarchical features from network traffic data, we could achieve higher classification accuracy, thereby enhancing the effectiveness of our malware detection system.

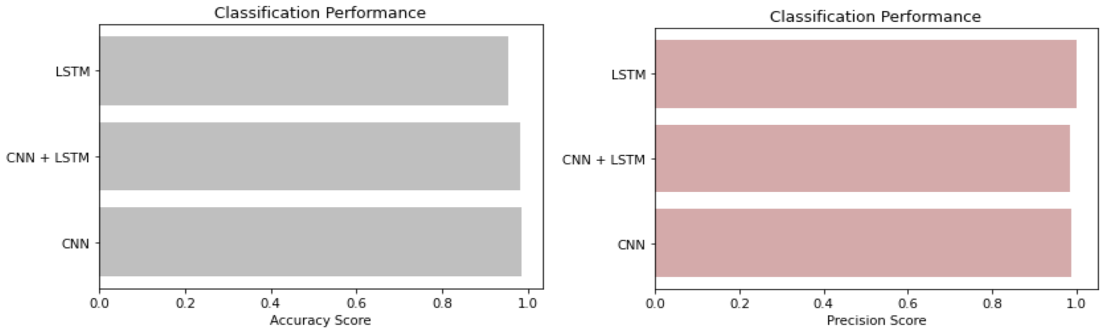


Fig 4: The comparison graph of Accuracy and Precision score of the models

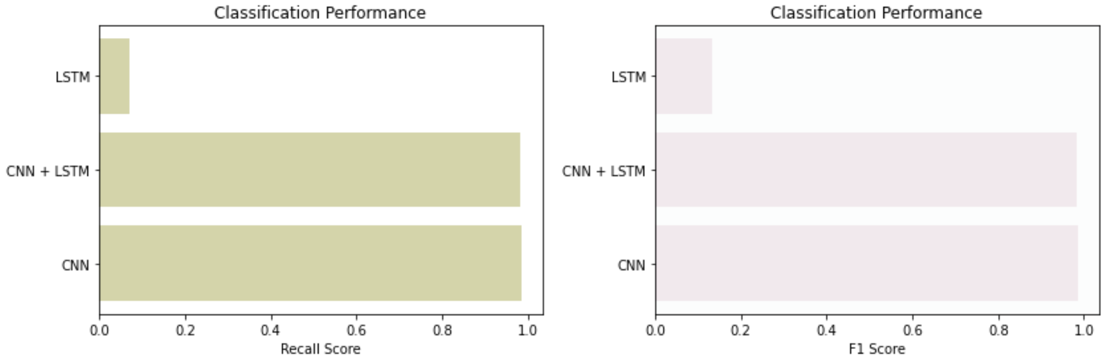


Fig 5: The comparison graph of Recall score and F1 Score of the models

VI. CONCLUSION

The project successfully demonstrated the effectiveness of a multitask LSTM model in detecting diverse IoT malware threats. Furthermore, the integration of LSTM networks enhanced the model's ability to analyze time-series data, uncovering intricate patterns within IoT network traffic. The addition of CNN and CNN+LSTM extensions further bolstered IoT security, with CNN exhibiting slight superiority and thus being deployed for reinforced protection against evolving malware threats. Future iterations aim to refine these models by incorporating more sophisticated deep learning architectures, optimizing real-time processing through edge computing techniques, and enhancing adaptability through dynamic threat intelligence feeds. Overall, the proposed multitask LSTM-based DL approach proved effective in classifying and identifying malware network traffic. Further improvements will involve utilizing additional datasets to generalize the model and integrating modules to strengthen protection and mitigation procedures, ultimately ensuring continuous and robust security in diverse IoT ecosystems.

REFERENCES

[1] H. N. Saha, A. Mandal, and A. Sinha, “Recent trends in the Internet of Things,” in Proc. IEEE 7th Annu. Comput. Commun. Workshop Conf. (CCWC), 2017, pp. 1–4.

[2] “2020 unit 42 IoT threat report.” Unit 42. Mar. 2020. Accessed: Apr. 17, 2022. [Online]. Available: https://start.paloaltonetworks.com/ unit-42-iot-threat-report

[3] M. Antonakakis et al., “Understanding the mirai botnet,” in Proc. 26th USENIX Security Symp. (USENIX Security), 2017, pp. 1093–1110.

[4] J. Vijayan. “Satori botnet malware now can infect even more IoT devices.” 2018. [Online]. Available: https://www.darkreading.com/vulnerabilities-threats/satori-botnet-malware-now-can-infect-evenmore-iotdevices

[5] C. Cimpanu et al., “Hajime botnet makes a comeback with massive scan for MikroTik routers.” 2018. [Online]. Available: https://www. radware.com/newsevents/mediacoverage/2018/hajime-botnet-makes-acomeback-with-massive-scan/

[6] L. Pascu. “78% of malware activity in 2018 driven by IoT botnets, NOKIA finds.” 2018. [Online]. Available: https://www.bitdefender.com/ blog/hotforsecurity/78-malware-activity-2018-driven-iot-botnets-nokiafinds

[7] P.-A. Vervier and Y. Shen, “Before toasters rise up: A view into the emerging IoT threat landscape,” in Proc. Int. Symp. Res. Attacks Intrusions Defenses, 2018, pp. 556–576.

[8] H. Haddadi, V. Christophides, R. Teixeira, K. Cho, S. Suzuki, and A. Perrig, “Siotome: An edge-ISP collaborative architecture for IoT security,” in Proc. IoTSec, 2018, pp. 1–4.

[9] T. Zixu, K. S. K. Liyanage, and M. Gurusamy, “Generative adversarial network and auto encoder based anomaly detection in distributed IoT networks,” in Proc. IEEE Global Commun. Conf. (GLOBECOM), 2020, pp. 1–7.

[10] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani, “Characterization of encrypted and VPN traffic using time-related features,” in Proc. 2nd Int. Conf. Inf. Syst. Security Privacy (ICISSP), 2016, pp. 407–414.

[11] R. Mills, A. K. Marnerides, M. Broadbent, and N. Race, “Practical intrusion detection of emerging threats,” IEEE Trans. Netw. Service Manag., vol. 19, no. 1, pp. 582–600, Mar. 2022.

[12] T. M. Booij, I. Chiscop, E. Meeuwissen, N. Moustafa, and F. T. H. den Hartog, “ToN_IoT: The role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion data sets,” IEEE

Internet Things J., vol. 9, no. 1, pp. 485–496, Jan. 2022.

[13] I. Ullah and Q. H. Mahmoud, “Network traffic flow based machine learning technique for IoT device identification,” in Proc. IEEE Int. Syst. Conf. (SysCon), 2021, pp. 1–8.

[14] Z. Chen et al., “Machine learning-enabled IoT security: Open issues and challenges under advanced persistent threats,” ACM Comput. Surv., to be published. [Online]. Available: https://doi.org/10.1145/3530812

[15] M. R. P. Santos, R. M. C. Andrade, D. G. Gomes, and A. C. Callado, “An efficient approach for device identification and traffic classification in IoT ecosystems,” in Proc. IEEE Symp. Comput. Commun. (ISCC), 2018, pp. 304–309.

[16] A. Sivanathan, H. H. Gharakheili, and V. Sivaraman, “Managing IoT cyber-security using programmable telemetry and machine learning,” IEEE Trans. Netw. Service Manag., vol. 17, no. 1, pp. 60–74, Mar. 2020.

[17] M. Alhanahnah, Q. Lin, Q. Yan, N. Zhang, and Z. Chen, “Efficient signature generation for classifying cross-architecture IoT malware,” in Proc. IEEE Conf. Commun. Netw. Security (CNS), 2018, pp. 1–9.

[18] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, “Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges,” IEEE Trans. Netw. Service Manag., vol. 16, no. 2, pp. 445–458, Jun. 2019.

[19] A. M. Sadeghzadeh, S. Shiravi, and R. Jalili, “Adversarial network traffic: Towards evaluating the robustness of deep-learning-based network traffic classification,” IEEE Trans. Netw. Service Manag., vol. 18, no. 2, pp. 1962–1976, Jun. 2021.

[20] A. H. Lashkari, G. Draper-Gil, M. S. I. Mamun, and A. A. Ghorbani, “Characterization of Tor traffic using time based features,” in Proc. ICISSp, 2017, pp. 253–262.

[21] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, “Network intrusion detection for IoT security based on learning techniques,” IEEE Commun. Surveys Tuts., vol. 21, no. 3, pp. 2671–2701, 3rd Quart., 2019. [22] R. Zhao. “NSL-KDD.” 2022. [Online]. Available: https://dx.doi.org/10. 21227/8rpg-qt98

[23] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” in Proc. IEEE Symp. Comput. Intell. Security Defense Appl., 2009, pp. 1–6.

[24] N. Moustafa, 2019, “UNSW_NB15 Dataset,” IEEE DataPort. [Online]. Available: https://dx.doi.org/10.21227/8vf7-s525

[25] S. Garcia, M. Grill, J. Stiborek, and A. Zunino, “An empirical comparison of botnet detection methods,” Comput. Security, vol. 45, pp. 100–123, Sep. 2014. [Online]. Available: https://doi.org/10.1016/j. cose.2014.05.011

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

☎ 9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details