Performance Analysis of LSTM, SVM, CNN, and CNN-LSTM Algorithms for Malware Detection in IoT Dataset

ILIYAN BARZEV, DANIELA BORISSOVA
Department of Information Processes and Decision Support Systems,
Institute of Information and Communication Technologies at the Bulgarian Academy of Sciences,
Acad. G. Bonchev Str. Bl. 2, Sofia 1113,
BULGARIA

Abstract: - Machine learning is an effective technique to tackle both the detection and classification tasks of malware. This is realized through learning algorithms that use various distinguishing features that characterize malware. Today's malware uses extremely sophisticated techniques, which means that various techniques to combat it are intensively developed. When malware is invisible, it can compromise many different data of a large number of users. Therefore, it is necessary to first analyze the types of malicious software and then propose appropriate countermeasures. In this regard, this work aims to analyze the performance of some well-known machine-learning techniques based on neural networks and support vector machines, originally developed as a method for the efficient training of neural networks. For the goal SVM, LSTM, CNN, and CNN-LSTM algorithms are analyzed concerning their effectiveness in the classification of malware in IoT datasets. For all the algorithms studied, their confusion matrices are presented along with receiver operating characteristic curves. The best results were obtained using the hybrid CNN-LSTM approach. Its results showed an accuracy of 97% and balanced performance across all metrics.

Key-Words: - machine learning, malware detection, performance analysis, LSTM, SVM, CNN, CNN-LSTM, IoT.

Received: May 24, 2024. Revised: December 23, 2024. Accepted: January 25, 2025. Published: April 14, 2025.

1 Introduction

Today, malware protection is important for several reasons. One concerns data security as malicious software could compromise personal and financial data, leading to identity theft or financial loss. The second is related to systems health because infections with viruses and Trojan horses can damage computers and networks, resulting in information loss and expensive repairs. Malware can also affect productivity by causing crashes disrupting day-to-day operations and losing time and resources. All of these situations are reflected in the companies' reputation – data compromise can impact customer trust which is difficult to recover. Last but not least is compliance with laws and regulations as many organizations are required by law to protect their customers' data. Violations can lead to serious penalties. Malware can spread from one device to another, endangering the entire network security.

Network security must limit external access in such a way as to ensure the confidentiality and integrity of data and resources. There is a constant threat of cyber-attacks due to the variety of applications of IoT devices in healthcare [1], digital economy [2], smart city [3], maritime industry [4], agriculture [5], etc. The IoT devices are used not only in the company where we work but in our homes too [6]. One of the critical challenges related to malware penetration is gaining unauthorized access to IoT devices, where the malware attempts to copy authorized devices by mimicking their hardware and software specifications, [7]. Once in a network, malware can infect the entire network and remain inactive for days or weeks. The best security programs not only scan for malware on login but also continuously monitor files afterward to detect anomalies and eliminate malware. The most effective way to protect is to use cyber ranges to conduct research and tests on real-world systems to discover their weaknesses, [8].

Companies are making efforts to protect themselves against network threats, but with the emergence of new malware, this is a constant challenge. To ensure reliable security, it is necessary to use both different hardware and software solutions, [9]. Therefore, it is important to take some measures, the simplest of which is the installation of antivirus software. All this motivates the authors of this paper to compare the

performance of some machine learning algorithms and their derivatives in detecting malware.

Machine learning (ML) is an essential element of malware detection techniques, and this is due to several main factors:

- *Data Analysis*: ML algorithms are able to analyze huge amounts of data and complex datasets to detect patterns and anomalies due to malware actions.
- *Classification*: ML algorithms are capable of detecting and classifying malicious software using trained models.
- *Behavioral Analysis*: Malware detection can be done by using behavioral analysis to observe and interact with a malware sample. Fragments of malicious code can be identified on static analysis or patterns.
- *Automatic Learning*: ML-based malware detection models can recognize new threats due to the ability to be trained with new threat information thus being able to identify changing new behaviors.
- *Threat Prediction:* ML techniques can be used to predict including threat intelligence anomaly detection. This is due to the fact that they can be trained on a variety of data, including historical data, based on which to look for behavioral trends.
- *Semantic Understanding*: Semantic analysis in ML involves building structures that approximate concepts from a large set of files. This makes it possible to extract valuable information from the context of files, which is important for malware detection.
- *Email Classification*: ML models contribute to analyzing email content thus allowing real-time detection of phishing emails and identifying suspicious patterns.

Increasing the ability to detect malware can be achieved through the use of various ML algorithms. Thus, using appropriate ML models will contribute to improving cybersecurity. ML is concerned with developing algorithms and models that allow computers to learn from data in order to improve their performance without explicit programming. ML requires large amounts of data from which algorithms can extract information and patterns. Models are trained using historical data. Thus trained models can predict or classify new data based on what they have learned. Over time, as new data becomes available, the models can be adapted and improved. A review of malware detection techniques and algorithms can be found in [10]. Malware analysis techniques can be classified as static, dynamic, hybrid, and memory analysis, [11]. Two of them are basic techniques for malware analysis – static and dynamic. Static analysis aims to examine the malware's code and structure without running the code, while dynamic analysis examines behavior during code execution, [12]. Many researchers try to propose different techniques to cope with the challenges of malware detection and analysis, [13], [14]. For example, authors proposed a framework for the detection of malware combining deep learning and ML, [15]. The malware detection effectiveness depends on how effectively distinguishing features of malware are extracted by analysis techniques. There are different methods for analysis using different static and dynamic tools as shown in [16], [17], [18].

Considering the challenging problem related to malware detection, the current article aims to analyze algorithms based on neural networks and Support Vector Machines (SVM), which were originally developed as a method for the efficient training of neural networks. The analyzed algorithms are Long Short-Term Memory Networks (LSTM), SVM, Convolutional Neural Network (CNN), and CNN-LSTM with respect to their effective classification of malware in IoT datasets.

2 Preliminary

2.1 Support Vector Machines

Support Vector Machines are a set of supervised learning methods that can analyze data for prediction and classification. The SVM algorithm aims to find such a hyperplane in an N-dimensional space that clearly can classify the data points, [19].

2.2 Long Short-Term Memory Networks

Long Short-Term Memory Networks can be expressed as a sequential neural network with the ability to persist the information over arbitrary time intervals. LSTM is a deep neural network capable of coping with information from time series, suitable for the prediction of long-term nonlinear series, [20]. The LSTM-based forecasting model extracts non-linear and dynamic features of the process data to achieve satisfactory forecasting performance.

2.3 Convolutional Neural Network

Convolutional Neural Networks as a type of deep learning algorithm often used to analyze visual imagery. The basic idea behind CNN is to use a series of convolutional layers to extract features

from the input, followed by one or more fully connected layers to produce the final prediction, [21]. CNN uses a special technique known as "convolution" and relies on matrix multiplications to combine two functions to show how one changes the shape of the other.

2.4 CNN-LSTM

The hybrid CNN-LSTM method combines the advantages of CNN for feature extraction, while LSTM contributes to achieving better classification results, [22], [23].

3 Basic Steps for Experiments Conducting

For this study, a standard methodology is used to analyze and classify network traffic data from the IoT-23 dataset. This dataset contains both benign and malicious network activities, making it a suitable benchmark for evaluating the effectiveness of machine learning algorithms. The article aims to compare the following ML models LSTM, SVM, CNN, and CNN-LSTM for malware detection, following the methodology basic steps shown in Figure 1.

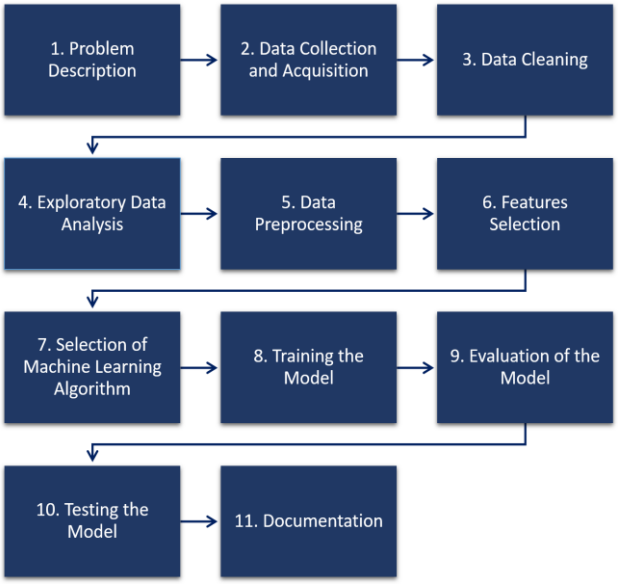

Fig. 1: Basic steps for conducting experiments

This generalized approach aims to provide a comprehensive understanding of the data, including selection of optimal model, and assessment of the performance. Each step of the methodology shown in Figure 1 is briefly described focusing on the processes and challenges as follows:

1) *Problem Description*. This paper aims to detect and classify malicious activities in IoT network traffic using the IoT-23 dataset. This is an initial step in setting the pace for the experiment by stating the objectives, scope, and intended results. The problem statement needs much accuracy since it is hard to frame it comprehensively, considering the context of the dataset, possible threats, and expected outcomes. In addition, there are multiple attack types in the IoT-23 dataset which requires a clear understanding of the dataset's context.
2) *Data Collection and Acquisition*. This step requires gathering the IoT-23 dataset network traffic data, which includes all benign and malicious activities of interest. The integrity of the data can be compromised, ensuring that it is complete. It may also be tricky to get access to IoT traffic data that is comprehensive and well-labeled. Ensuring that the downloaded data from the IoT-23 data is vital, because some issues may be seen, such as corrupted packet capture (PCAP) files, [24].
3) *Data Cleaning*. The process of data cleaning is needed to remove the noise, handle missing values, and deal with inconsistencies within the dataset to enhance the quality of the data. Data cleaning is a prerequisite for the dataset to be subjected to analysis and modeling. The cleaning process is a mandatory step, as it is possible for the dataset to contain corrupted data, duplicate records, or missing values. Therefore, proper cleaning is very important and errors made can negatively affect the performance of the model. For this dataset, the Wireshark tool was used, which helps identify and filter out corrupted packets or incomplete sessions in the packet capture files, which represent the data units and payload that make up the network traffic.
4) *Exploratory Data Analysis*. This analysis is done to identify the hidden structure and distribution of the data. For this purpose, data visualization techniques, descriptive statistics, and feature correlation analysis are used, which can improve the identification of patterns and potential anomalies. For the analysis of complex data such as network traffic, the use of a large number of features is difficult. Significant efforts are required to identify significant patterns in the presence of noise and outliers. It should be noted that in the considered IoT-23 dataset, outliers were found that require more careful analysis for pattern recognition, [25].

5) *Data Preprocessing*. This preprocessing of data aims to transform it into a format suitable for machine learning. Feature encoding, normalization, and dimensionality reduction can be used at this stage. The choice of specific techniques for preprocessing network traffic data depends on various factors. For example, in noise reduction, it is important to preserve temporal relationships.
6) *Feature Selection*. Feature selection aims to find the highly relevant features contributing to the predictive power of the model, [26]. This is reflected in reducing the dimensionality, improving model interpretability, and minimizing overfitting. It is not straightforward to choose the optimal subset of features since feature relevance and redundancy have to be balanced. Overlooking critical features will degrade the model's performance, [27]. The use of the Python library Featurewiz was helpful in selecting appropriate features from the IoT-23 dataset.
7) *Selection of Machine Learning Algorithm*. This step is where one selects the most appropriate machine learning algorithms for the task at hand. For this research, LSTM, SVM, CNN, and CNN-LSTM models were selected to uncover different characteristics and patterns of the IoT-23 dataset. Finding the best-suited algorithm depends on understanding the dataset well and understanding the strengths of each model. Ensuring that the chosen models align with the data's characteristics is critical for optimal performance. Understanding the nature of the dataset's characteristics is crucial to align the different ML models with their capabilities for optimal performance, and choosing a model that is not so appropriate to our experiment, so that we can have a better view of the different results.
8) *Training the Model*. To train the selected models, a pre-processed dataset is required. The data is fed into the selected training models, accompanied by hyperparameter tuning and minimization of the loss functions through iteration. The process of training the model(s) can be quite time-consuming for complex models such as CNN-LSTM. Avoiding overfitting and ensuring convergence are serious challenges at this stage.
9) *Evaluation of the Model*. Accuracy, precision, recall, and F1-score will be used to evaluate the performance of the ML models. These will be the metrics against which the effectiveness of the trained models in distinguishing between benign and malicious network traffic will be determined. Complex data such as those provided by the IoT-23 dataset require an efficient evaluation of the model, probably with problems related to data balancing and effective validation techniques.
10) *Testing the Model*. The models are applied to unseen data to measure their generalization performance in real-world settings. Testing on new data may expose model bias, overfitting, or failure to generalize. It is crucial that the performance of the model translates effectively to different environments.
11) *Documentation*. This is the final step and involves documentation of all the activities undertaken, from data pre-processing steps to model configurations to results, including challenges posed during the experiment. Clear documentation requires attention to detail, although it is time-consuming, it is necessary in research work.

Following the methodology described above, it is possible to analyze IoT network traffic data. Each step of the methodology provides a transparent workflow that leads to a clear view of IoT security threats. The combination of LSTM, SVM, CNN, and CNN-LSTM will be used to capture various patterns in the data, improving the detection and classification of malicious activities.

4 Result Analysis and Discussion

The performance of each classification algorithm is determined by the corresponding confusion matrix. In particular, for the investigated IoT-23 dataset, the performance of LSTM, SVM, CNN, and CNN-LSTM are shown in Figure 2, Figure 3, Figure 4, and Figure 5, which visualize the performance of individual classification algorithms.

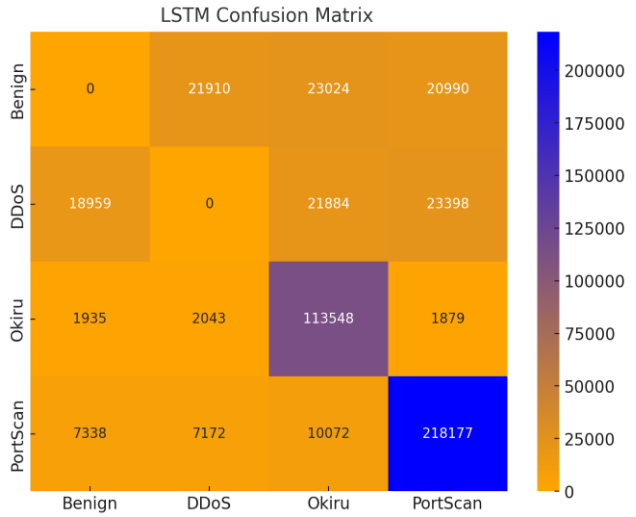

Fig. 2: Confusion matrix of LSTM

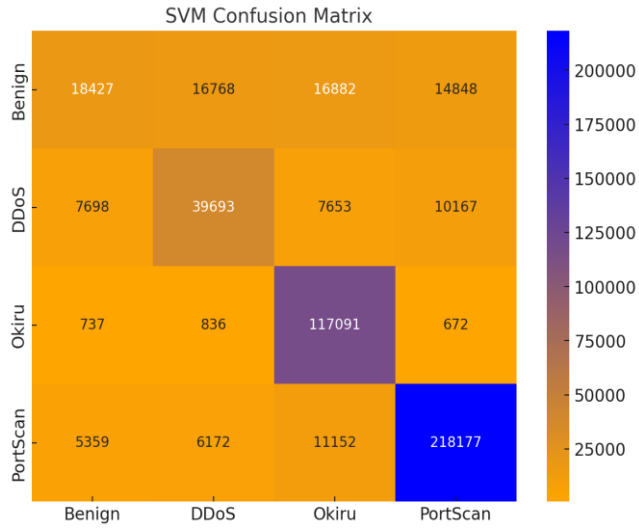

Fig. 3: Confusion matrix of SVM

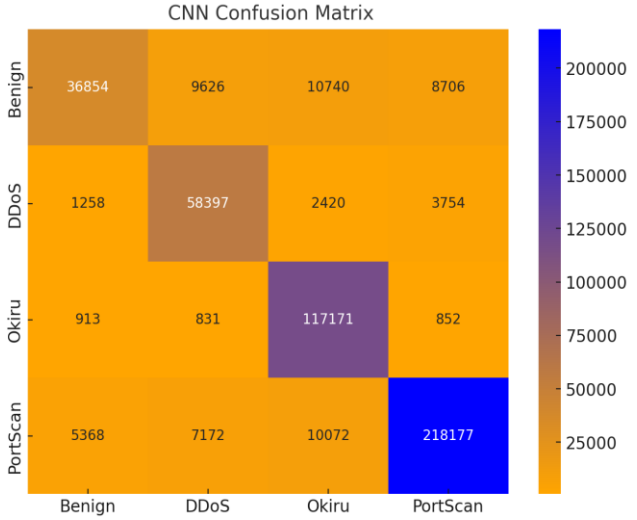

Fig. 4: Confusion matrix of CNN

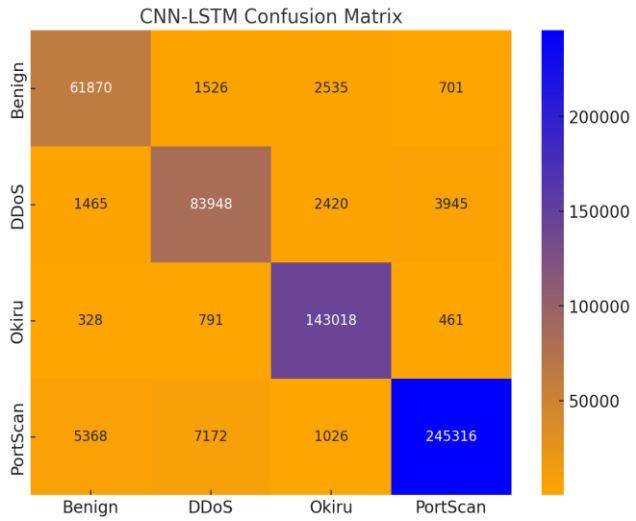

Fig. 5: Confusion matrix of CNN-LSTM

To evaluate the performance of the studied algorithms (LSTM, SVM, CNN, and CNN-LSTM) the receiver operating characteristic (ROC) curve was used. These graphs (ROC curves) illustrate the discrimination ability of each model by expressing the true positive rate (TPR) versus the false positive rate (FPR) for different threshold values as shown in Figure 6 (Appendix).

Comparing these ROC curves for each model allows us to assess the trade-offs between sensitivity and specificity. Thus, analysis of these graphs can be used to determine how well each algorithm identifies positive cases while minimizing false positives. Consequently, the overall classification performance at different thresholds can be determined for the respective algorithms being compared as can be seen from Figure 6 (Appendix).

A higher curve with a larger area under the curve (AUC) suggests better performance in classifying instances for that specific class. The shape and placement of each curve reflect the trade-offs between correctly identifying positive instances and avoiding false positives.

Based on performance metrics it is possible to determine the particular ability of each algorithm to classify correctly while minimizing false positives and balancing between precision and recall.

The comparison of the effectiveness of the four algorithms subject to the current study, analyzed by four key indicators like accuracy, precision, recall, and F1-score is shown in Figure 7 (Appendix).

From Figure 7 (Appendix) can be seen that the LSTM algorithm demonstrated the lowest value for accuracy of 0.74 also for precision (0.72), recall (0.74), and F1-score (0.71). LSTM shows capability to capture temporal patterns inherent in network traffic data, but its performance was insufficient

toward precision and sensitivity compared to other models for this dataset.

The second algorithm with better performance results is SVM. Its overall result is equal to 3.88. The ability of SVM to separate classes by a hyperplane worked effectively for the considered IoT-23 dataset. This algorithm is limited by its lack of deep feature extraction capabilities, making it less suitable for highly complex relationships in network traffic data.

The next algorithm with better performance than LSTM and SVM is CNN. It has a value of 0.92 for precision and 0.91 for the rest of the metrics (see Figure 7, Appendix). As CNN is capable of extracting spatial features from network traffic patterns, the results for the IoT-23 dataset show much better classification malware detection compared to the LSTM and SVM.

The results for the hybrid CNN-LSTM algorithm are the best compared to the rest algorithms. It achieves a value of 0.97 for all metrics – accuracy, precision, recall, and F1-score. This is due to its combining strengths of CNN's feature extraction with the ability of LSTM to capture temporal dependencies. The IoT-23 dataset contains both spatial and temporal patterns in network traffic behavior, and these specifics make the CNN-LSTM algorithm making it especially effective at capturing these complex interactions.

All evaluated models demonstrate levels of success when applied to the IoT-23 dataset, but the hybrid CNN-LSTM approach gives us the highest accuracy and strong performance on every metric. Its ability to combine efficient feature extraction with sequential learning makes it particularly suitable for dealing with the complex nature of IoT network traffic data in malware detection tasks. However, practical factors such as computational cost and resource allocation must be carefully considered before integrating this method into real-world systems.

5 Conclusion

Malware detection is a challenging task and that is why the current article compares some ML algorithms about their classification. This study compares the effectiveness of the classification of some algorithms based on neural networks such as CNN, LSTM, and CNN-LSTM, and SVM originally developed as a method for efficient training of neural networks. These algorithms are compared with respect to four metrics accuracy, precision, recall, and F1-score. The results show that the hybrid CNN-LSTM approach is the best among other algorithms with a value of 0.97 for all metrics. This is due to combining the strengths of CNN's feature extraction with the ability of LSTM to capture temporal dependencies. Considering these results we plan future experiments with other combinations of well-known algorithms such as logistic regression and partial least squares for future research.

Acknowledgment:
This work is supported by the Bulgarian National Science Fund by the project "Mathematical models, methods and algorithms for solving hard optimization problems to achieve high security in communications and better economic sustainability", KP-06-H52/7/19-11-2021 and is supported also by the Bulgarian National Science Fund by the project "Innovative Methods and Algorithms for Detection and Recognition of Moving Objects by Integration of Heterogeneous Data", KP-06-N 72/4/05.12.2023.

Declaration of Generative AI and AI-assisted Technologies in the Writing Process

During the preparation of this work the authors used ChatGPT 4o mini in order to summarize the main advantages of ML that contributes to malware detection. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References:

[1] K.H. Almotairi, Application of internet of things in healthcare domain, *Journal of Umm Al-Qura University for Engineering and Architecture*, Vol. 14, 2023, pp. 1–12, https://doi.org/10.1007/s43995-022-00008-8.

[2] A. Khang, V. Abdullayev, V. Hahanov, V. Shah, (Eds.). Advanced IoT Technologies and Applications in the Industry 4.0 Digital Economy, *CRC Press*. 2024, https://doi.org/10.1201/9781003434269.

[3] M.E.E Alahi, A. Sukkuea, F.W. Tina, A. Nag, W. Kurdthongmee, K. Suwannarat, S.C. Mukhopadhyay, Integration of IoT-enabled technologies and artificial intelligence (AI) for smart city scenario: Recent advancements and future trends, *Sensors*. Vol. 23(11), 2023, 5206, https://doi.org/10.3390/s23115206.

[4] S. Aslam, H. Herodotou; E. Garro, Á. Martínez-Romero, M.A. Burgos, A. Cassera, G. Papas, P. Dias, M.P. Michaelides, IoT for

the maritime industry: Challenges and emerging applications, In: *2023 18th Conference on Computer Science and Intelligence Systems* (FedCSIS), Warsaw, Poland, 2023, pp. 855-858, https://doi.org/10.15439/2023F3625.

[5] S. K. Smmarwar, G.P. Gupta, S. Kumar, Deep malware detection framework for IoT-based smart agriculture, *Computers and Electrical Engineering*, Vol. 104, Part A, 108410, 2022, https://doi.org/10.1016/j.compeleceng.2022.108410.

[6] I. Cvitić, D. Peraković, M. Periša, A. Jevremović, A. Shalaginov, An overview of smart home IoT trends and related cybersecurity challenges, *Mobile Networks and Applications*, Vol. 28, 2023, pp. 1334–1348. https://doi.org/10.1007/s11036-022-02055-w.

[7] E. Praveen Kumar, S. Priyanka, A comprehensive survey on hardware-assisted malware analysis and primitive techniques, *Computer Networks*, Vol. 235, 2023, 109967, https://doi.org/10.1016/j.comnet.2023.109967.

[8] I. Blagoev, V. Shalamanov, Development of Cyber Ranges as a reference environment for digital transformation, In: *2023 4th International Conference on Communications, Information, Electronic and Energy Systems* (CIEES), Plovdiv, Bulgaria, 2023, pp. 1-5, https://doi.org/10.1109/CIEES58940.2023.10378806.

[9] A. Mansouri, A. Elzaar, M. Madani, T. Bakir, Design and Hardware Implementation of CNN-GCN Model for Skeleton-Based Human Action Recognition, *WSEAS Transactions on Computer Research*, Vol. 12, 2024, pp. 318-327, https://doi.org/10.37394/232018.2024.12.31.

[10] Ö.A. Aslan, R. Samet, A comprehensive review on malware detection approaches, In: *IEEE Access*, Vol. 8, 2020, pp. 6249-6271, https://doi.org/10.1109/ACCESS.2019.2963724.

[11] R. Sihwail, K. Omar, K.A.Z. Ariffin, A survey on malware analysis techniques: Static, dynamic, hybrid and memory analysis, *International Journal on Advanced Science, Engineering and Information Technology*, Vol. 8(4-2), 2018, pp. 1662-1671, https://doi.org/10.18517/ijaseit.8.4-2.6827.

[12] C. Raghuraman, S. Suresh, S. Shivshankar, R. Chapaneri, Static and dynamic malware analysis using machine learning, In: Luhach, A., Kosa, J., Poonia, R., Gao, XZ., Singh, D. (eds) *First International Conference on Sustainable Technologies for Computational Intelligence. Advances in Intelligent Systems and Computing*, Vol 1045, 2020, https://doi.org/10.1007/978-981-15-0029-9_62.

[13] M. Drolet, V. Roberge, Enterprise Malware detection using digital forensic artifacts and machine learning, *WSEAS Transactions on Computer Research*, Vol. 12, 2024, pp. 336-347, https://doi.org/10.37394/232018.2024.12.33.

[14] M. Nobakht, R. Javidan, A. Pourebrahimi, SIM-FED: Secure IoT malware detection model with federated learning, *Computers and Electrical Engineering*, Vol. 116, 2024, 109139, https://doi.org/10.1016/j.compeleceng.2024.109139.

[15] K. Shaukat, S. Luo, V. Varadharajan, A novel deep learning-based approach for malware detection, *Engineering Applications of Artificial Intelligence*, Vol. 122, 2023, 106030, https://doi.org/10.1016/j.engappai.2023.106030.

[16] J. Singh, and J. Singh, A survey on machine learning-based malware detection in executable files, *Journal of Systems Architecture*, Vol. 112, 2021, 101861, https://doi.org/10.1016/j.sysarc.2020.101861.

[17] M. Gopinath, S. Ch. Sethuraman, A comprehensive survey on deep learning based malware detection techniques, *Computer Science Review*, Vol. 47, 2023, 100529, https://doi.org/10.1016/j.cosrev.2022.100529.

[18] X. Ling, L. Wu, J. Zhang, Z. Qu, W. Deng, X. Chen, Y. Qian, C. Wu, S. Ji, T. Luo, J. Wu, Y. Wu, Adversarial attacks against Windows PE malware detection: A survey of the state-of-the-art, *Computers & Security*, Vol. 128, 2023, 103134, https://doi.org/10.1016/j.cose.2023.103134.

[19] S. Suthaharan, Support Vector Machine. In: Machine Learning Models and Algorithms for Big Data Classification, *Integrated Series in Information Systems,* vol. 36, 2016, pp. 207-235, https://doi.org/10.1007/978-1-4899-7641-3_9.

[20] S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Computation*, Vol. 9(8), 1997, pp. 1735-1780, https://doi.org/10.1162/neco.1997.9.8.1735.

[21] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G.

Wang, J. Cai, T. Chen, Recent advances in convolutional neural networks, *Pattern Recognition*, Vol. 77, 2018, pp. 354-377, https://doi.org/10.1016/j.patcog.2017.10.013.

[22] H. D. Shoorkand, M. Nourelfath, A. Hajji, A hybrid CNN-LSTM model for joint optimization of production and imperfect predictive maintenance planning, *Reliability Engineering & System Safety*, Vol. 241, 2024, 109707, https://doi.org/10.1016/j.ress.2023.109707.

[23] F. O. Ozkok, M. Celik, A hybrid CNN-LSTM model for high resolution melting curve classification, *Biomedical Signal Processing and Control*, Vol. 71, Part A, 2022, 103168, https://doi.org/10.1016/j.bspc.2021.103168.

[24] S. Garcia, A. Parmisano, M. J. Erquiaga. IoT-23: A labeled dataset with malicious and benign IoT network traffic (Version 1.0.0) [Data set], 2020, Zenodo. http://doi.org/10.5281/zenodo.4743746.

[25] A. Sharma and H. Babbar, Understanding IoT-23 Dataset: A Benchmark for IoT Security Analysis, In: *4th International Conference on Intelligent Technologies (CONIT)*, Bangalore, India, 2024, pp. 1-5, https://doi.org/10.1109/CONIT61985.2024.10627334.

[26] R. Ranjan, J. K. Chhabra, A Modified binary arithmetic optimization algorithm for feature selection, *WSEAS Transactions on Computer Research*, Vol. 11, 2023, pp. 199-205, https://doi.org/10.37394/232018.2023.11.18.

[27] M. Douiba, S. Benkirane, A. Guezzaz, M. Azrour. An improved anomaly detection model for IoT security using decision tree and gradient boosting. *The Journal of Supercomputing, Vol.* 79, 2023, pp. 3392-3411, https://doi.org/10.1007/s11227-022-04783-y.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed to the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.

APPENDIX

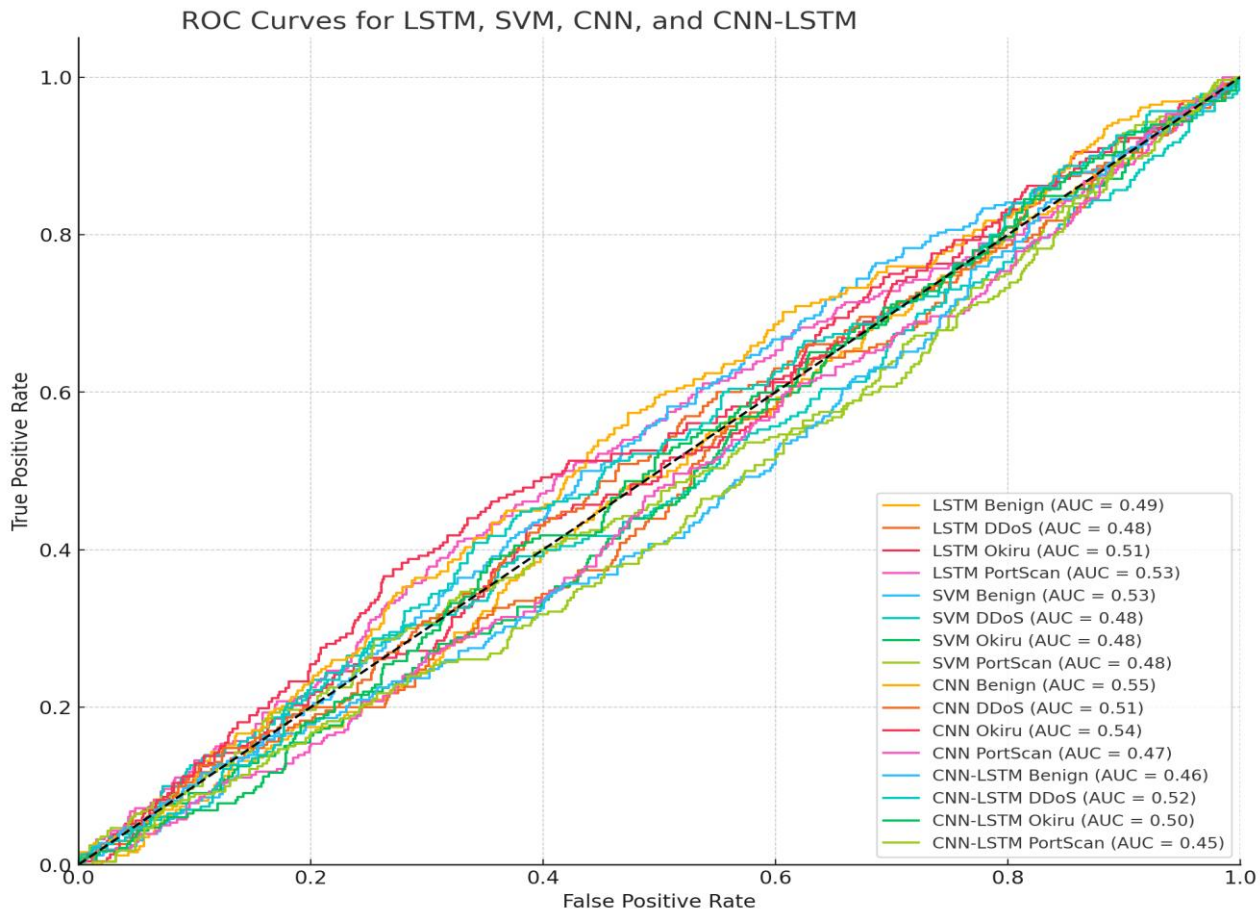

Fig. 6: ROC Curves for LSTM, SVM, CNN, CNN-LSTM

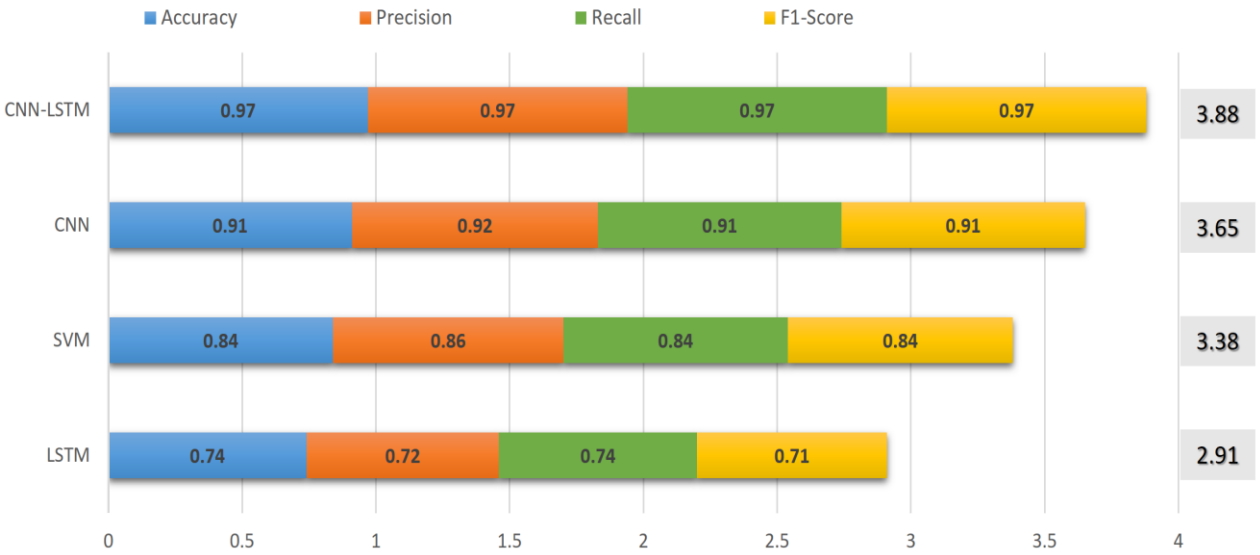

Fig. 7: Performance comparison between CNN-LSTM, CNN, SVM, and LSTM