2025 IEEE 14th International Conference on Communication Systems and Network Technologies (CSNT) | 979-8-3315-3193-5/25/$31.00 ©2025 IEEE | DOI: 10.1109/CSNT64827.2025.10968061

# Optimized Approaches to Malware Detection: A Study of Machine Learning and Deep Learning Techniques

Abrar Fahim
Department of Electrical and Electronic Engineering
Islamic University of Technology
Dhaka, Bangladesh
abrarfahim8@iut-dhaka.edu

Shamik Dey
Department of Electrical and Electronic Engineering
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
shamikdey7@gmail.com

Md. Nurul Absur
Department of Computer Science
City University of New York
New York, USA
mabsur@gradcenter.cuny.edu

Md Kamrul Siam
Department of Computer Science
New York Institute of Technology
New York, New York, USA
ksiam01@nyit.edu

Md. Tahmidul Huque
Department of Computer Science and Engineering
Bangladesh University of Business and Technology
Dhaka, Bangladesh
md.tahmidulhuque@gmail.com

Jafreen Jafor Godhuli
Department of Computer Science and Engineering
Bangladesh University of Business and Technology
Dhaka, Bangladesh
jafreenjaforgodhuli@gmail.com

***Abstract*—Digital systems find it challenging to keep up with cybersecurity threats. The daily emergence of more than 560,000 new malware strains poses significant hazards to the digital ecosystem. The traditional malware detection methods fail to operate properly and yield high false positive rates with low accuracy of the protection system. This study explores the ways in which malware can be detected using these machine learning (ML) and deep learning (DL) approaches to address those shortcomings. This study also includes a systematic comparison of the performance of some of the widely used ML models, such as random forest, multi-layer perceptron (MLP), and deep neural network (DNN), for determining the effectiveness of the domain of modern malware threat systems. We use a considerable-sized database from Kaggle, which has undergone optimized feature selection and preprocessing to improve model performance. Our finding suggests that the DNN model outperformed the other traditional models with the highest training accuracy of 99.92% and an almost perfect AUC score. Furthermore, the feature selection and preprocessing can help improve the capabilities of detection. This research makes an important contribution by analyzing the performance of the model on the performance metrics and providing insight into the effectiveness of the advanced detection techniques to build more robust and more reliable cybersecurity solutions against the growing malware threats.**

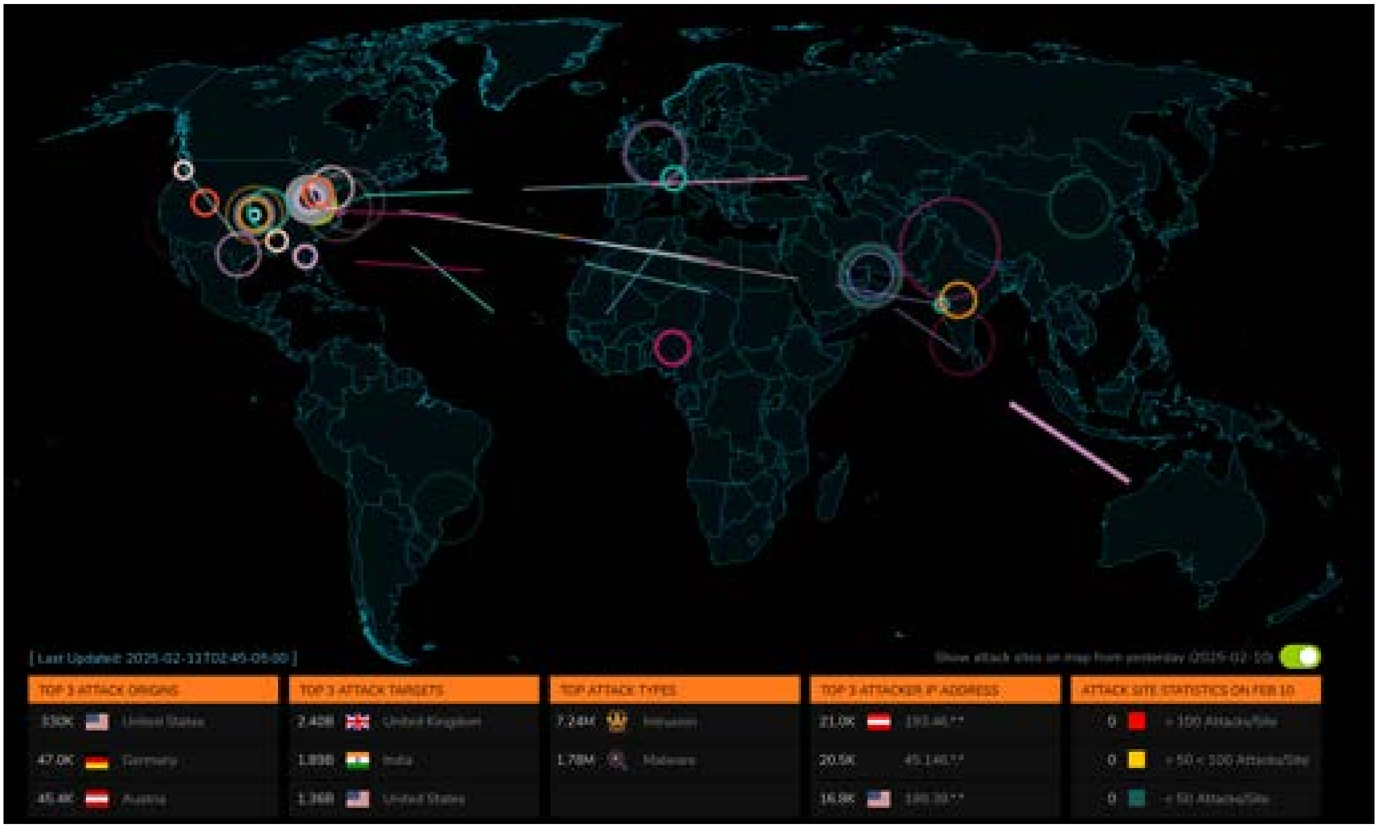
***Index Terms*—Malware Detection, Machine Learning, Deep Learning, Cybersecurity, Intrusion Detection, Feature Selection, Feature Extraction, Threat Detection, Cybercrime.**

## I. Introduction

According to Forbes and cybersecurity industry leaders, daily new malware variants were over 56,000 and this is enhancing the threat level to the world's cybersecurity systems [1]–[3]. Nevertheless, the dynamics of these threats continue to increase, whereas traditional approaches to malware identification can provide neither sufficient accuracy nor flexibility [4], [5], [17]. The increased vulnerability of sensitive data due to viruses, cyberattacks, cyberthreats, and cybercrimes must be addressed. The countries that encountered various attack types during a random week in 2025 are shown in Figure 1, along with the main malware suppliers.

Most current solutions have relatively low accuracy connected with a high number of false positives and do not allow efficient work with new, complex forms of malware. This creates a significant gap in the possibility of protection through detection since the current systems fail to recognize the modern threats that endanger the data and privacy of the people.However, the current level of ML and DL technologies has shown that malware detection has to be improved in order to fill this research gap. Due to the hardware improvements, the current ML algorithms can easily process large volumes of data and are suitable to detect any type of malware; however, they have their shortcomings, especially when implementing high-dimensional data sets or new strains of malware. In terms of features, DL models are capable of capturing and performing subtle details; hence, they can be applied to discover more enhanced malware. Moreover, there has not been sufficient effort made to compare the effectiveness of these techniques, which creates the need for a systematic approach.This research

979-8-3315-3193-5/25/$31.00 ©2025 IEEE
DOI: 10.1109/CSNT.2025.46



[ Last Updated: 2025-02-11T02:45-05:00 ]

Show attack sites on map from yesterday (2025-02-10)

| TOP 3 ATTACK ORIGINS | TOP 3 ATTACK TARGETS | TOP ATTACK TYPES | TOP 3 ATTACKER IP ADDRESS | ATTACK SITE STATISTICS ON FEB 10 |
|---|---|---|---|---|
| 330K United States | 2.40B United Kingdom | 7.24M Intrusion | 21.0K 193.46.*.* | 0 🟥 > 100 Attacks/Site |
| 47.0K Germany | 1.89B India | 1.78M Malware | 20.5K 45.141.*.* | 0 🟨 > 50 < 100 Attacks/Site |
| 45.4K Austria | 1.36B United States | | 16.8K 188.30.*.* | 0 🟩 < 50 Attacks/Site |

Fig. 1. Attacks throughout the world [6].

seeks to fill this gap by making the following two comparisons: (a) an assessment of the performance of different ML models for malware detection and, (b) an assessment of the performance of different DL models for malware detection. In this study, various algorithms, including Random Forest, MLP, and DNN, are evaluated to determine the best ways of enhancing the accuracy of malware detection with a focus on the modern threats.

## II. Related Works

There has been much research conducted and is being conducted in the area of malware detection through the use of machine learning and deep learning. Also, depending on the type of malware, researchers have also explored the performance of various models and datasets for malware detection, which produced various results. The following summaries highlight key findings from notable works in this area:

A recent study used a dataset from the Canadian Institute for Cybersecurity with 17,394 instances to assess the efficacy of three disparate algorithms: CNN, DT, and SVM with respect to the identification of malware [7]. Out of all the tested models, the Decision Tree model had the highest accuracy of 99%, whereby it clearly exhibits the capability of managing the dataset's feature space. The CNN was also impressive, achieving 98.76% accuracy; this could be attributed to the network's capability to identify tiered structures from data. Although its performance was not as high as the CNN with an accuracy of 96.41%, it was still feasible for use in malware classification.

In another study, the authors used a large number of machine learning methods to identify the Windows malware sample by using Kaggle data [8]. Among all the tested models, Random Forest was the best one, using which an overall accuracy of the model was found to be 99.44%. As for other models, SVM Decision Tree, ADA Boost, and all these models were also considered but did not appear to work well. The study also showed that Random Forest has other strengths, including the ensemble of decision trees to minimize overfitting.

DIVAKARLA et al. combined a new deep learning technique for Windows malware detection employing the EMBER dataset [9]. Their approach indeed got an accuracy of 87.76%, which shows that their approach is promising in handling polymorphic malware. A form of malware that morphs its code to avoid being detected is quite a nightmare to current detection techniques. The authors indicate that the application of deep learning methods may yield higher performance rates while analyzing intricate manifestations of malware activity; however, improvement is still possible.

Stacked ensemble learning was utilized by a group of researchers to conduct malware classification from the Portable Executable (PE) malware dataset of 19,611 samples [10]. This approach outperformed by combining predictions from multiple models. The accuracy of the Random Forest model was at the highest of 99.24%, closely followed by the Decision Tree model, which got 98.29%. This emphasizes the possibility of ensemble learning techniques used to enhance the strength of different classifiers in a composite detection process.

Machine learning models such as Random Forest and Gaussian Naive Bayes were explored by the author for the task of dynamic malware detection [11]. All training on wide PE files was taken from Kaggle. To the extent these results could be reproduced, they were achieved with an accuracy of 100% in behavior-based detection, showing that dynamic analysis, observing malware behavior during execution, is essential in overcoming the shortcomings of static features.

Another remarkable study presented a Hybrid Feature Malware Detection System (HF-MDS), which is built from the LSVC model by fusing different types of analysis, such as static and dynamic analysis [12]. The real-world malware precision achieved with this hybrid approach was 99.743%, showing that integrating multiple feature sets improves detection accuracy. It was found in the study that combining static and dynamic features can capture a wider range of malware characteristics with greater performance. Strategies like as ML has a track record of positively influencing society and improving it [13].

Furthermore, in another work, hardware performance counters (HPCs) were used to analyze malware detection, where MLP, CNN, and Full Order Radial Basis Function (RBF) were applied [14]. These models were able to report performance numbers as 96.95%, 98.2%, and 98.68%. HPC-based detection methods using low-level hardware data to detect malicious activity were highlighted as potential. By comparison, the performance of the CNN model demonstrates that it has the capacity to handle the spatial patterns behind HPC data.

## III. METHODOLOGY

### A. Data Collection

This study utilized the Malware Detection dataset, which was acquired from Kaggle, a well-known platform for data science and machine learning contests [16]. This dataset focuses on PC malware detection and comprises of 100,000 data points, each characterized by 35 distinct features. The dataset includes instances classified into two categories: 'benign' and 'malware'. The features capture various attributes relevant to network traffic and malware behavior.

### B. Preprocessing

Preprocessing has been defined as a critical component in data analysis and machine learning because it conditions or prepares the data so that we get the data in a proper format to influence the performance of the model [15].

- **Object Column Encoding**: Two object-type columns were encoded using *Label Encoding*, converting categorical values into numerical format for model compatibility.
- **Hash Column Unhashing**: A hashed column was unhashed using the `hashlib` library to retrieve the original data.
- **Outlier Removal Using Z-Score**: Outliers, defined as data points with Z-scores greater than 3 or less than -3, were removed to ensure the model was not affected by extreme values.
- **Feature Selection (Top 25 Features)**: *Recursive Feature Elimination (RFE)* was used to select the top 25 features for model training. Below table I shows the list of selected features.

TABLE I
TOP 25 SELECTED FEATURES

| Features | |
|---|---|
| millisecond | total_vm |
| state | shared_vm |
| usage_counter | exec_vm |
| prio | reserved_vm |
| static_prio | nr_ptes |
| normal_prio | end_data |
| policy | last_interval |
| vm_pgoff | nvcsw |
| vm_truncate_count | nivcsw |
| task_size | min_flt |
| cached_hole_size | maj_flt |
| free_area_cache | fs_excl_counter |
| mm_users | |

- **Train-Test Split (80-20)**: The dataset was split into training and testing sets, with 80% of the data (80,000 records) used for training and 20% (20,000 records) reserved for testing.

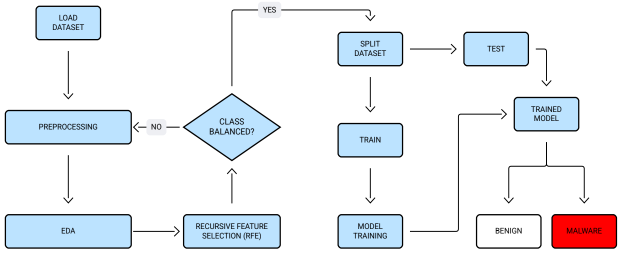


Fig. 2. : Workflow Diagram

Figure 2 shows the workflow of the expirement, that how malware detection is processed from beginning to end.

## IV. RESULTS AND DISCUSSION

To evaluate the performance of our models in predicting malware, we used accuracy, precision, recall, F1 score, AUC and confusion matrix as key metrics. The measure of accuracy reflects the total correctness of the model while the precision and recall measures the model's ability to find the real positive and the degree of reliability of the positive prediction respectively. This way we have a good quality measurement and the F1 score balances precision and recall. The AUC curve measures some of the discrimination capability of the model between classes, with higher values meaning better discrimination. The confusion matrix also shows a detailed breakdown of true and false classifications as well. The results illustrate the ability of our method in malware detection, together with the merit of feature selection and the proper preprocessing in upping the model performance.

In this section we present the result of many machine learning and deep learning models in detecting malware assessment based on classification metrics.

### A. Machine Learning Approach

The performance and training times of several machine learning models for malware detection are given in table II. with a highest final training accuracy of 99.77%, 10-fold CV accuracy of 99.67%, and a training time of 13.5 seconds the MLPClassifier was the best. With a final training accuracy of 99.97% and on a 10 fold CV accuracy of 98.45% it took 15.3 seconds for the KNeighborsClassifier to train. Logistic Regression had poor accuracy 93.75%; however, it was the fastest with 12.12 seconds of training. For 10-fold CV, with a final training accuracy of 99.72%, 99.64% for 10-fold CV, it was very good while requiring training time of 19.6s. Final training accuracy of Random Forest was 99.60%, CV accuracy of 99.52% with training times of 16.3 seconds. Training times for all models were in the range of 12 to 19.6 seconds and with high accuracy.

Summary of the testing accuracies and AUC-ROC values of the machine learning models is presented in the following table III for clearer classification performance of these models.

TABLE II
MACHINE LEARNING MODEL ACCURACIES AND TRAINING TIME

| Model | Train Accuracy (%) | CV Accuracy (%) | Train Time (in Sec.) |
|---|---|---|---|
| MLP | 99.77 | 99.67 | 13.5 |
| KNN | 99.97 | 98.45 | 15.3 |
| LR | 93.75 | 92.89 | 12.12 |
| SVC | 99.72 | 99.64 | 19.6 |
| RF | 99.60 | 99.52 | 16.3 |

This table III shows malware detection machine learning model results through test accuracy and AUC score measurements. Both Multi-Layer Perceptron (MLP) and Random Forest models demonstrated excellent performance with test accuracy reaching 99.99%. They maintained perfect scores of 100% for AUC. The Support Vector Classifier produced a test accuracy of 99.97% and reached a perfect AUC rating of 100%. The K-Nearest Neighbors model showed test accuracy of 99.97% and perfect AUC ranking at 100%. Logistic Regression demonstrated good results with a test accuracy of 93.75% and AUC score of 98.57% yet smaller than other approaches. The experiments show that all examined models work well for malware detection with their high accuracy scores and AUC values.

TABLE III
MODEL PERFORMANCE METRICS

| Model | Test Accuracy(%) | AUC Score(%) |
|---|---|---|
| Random Forest | 99.98 | 99.8 |
| SVC | 99.97 | 99.5 |
| Logistic Regression | 93.75 | 99 |
| K-Nearest Neighbors | 99.97 | 99.3 |
| Multi-Layer Perceptron | 99.99 | 99.6 |

To validate the results of the model Mcc and Kappa test is perfromed and the results are mentioned below in Table IV.

The table IV contains MCC and Kappa values for multiple models. The Random Forest SVC MLP and KNN classifiers deliver outstanding performance according to very high MCC and Kappa figures close to 100%. Logistic Regression produces results that perform well although not as highly as other models.

TABLE IV
MCC AND KAPPA VALUES FOR DIFFERENT MODELS

| Model | MCC (%) | Kappa (%) |
|---|---|---|
| Logistic Regression | 87.51 | 87.50 |
| Random Forest | 99.94 | 99.94 |
| Multi-Layer Perceptron (MLP) | 99.99 | 99.90 |
| K-Nearest Neighbors (KNN) | 99.94 | 99.94 |
| Support Vector Classifier (SVC) | 99.95 | 99.95 |

Figure 3 and 4 illustrate the confusion matrix and roc-curve of machine learning models respectively.

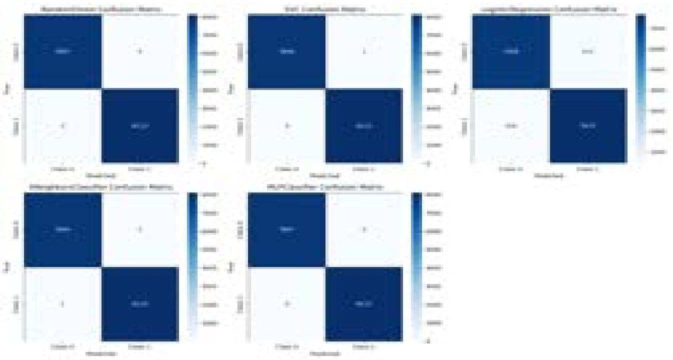


Fig. 3. Confusion Matrix

The figure 3 suggests, the most reliable models are Random Forest (RF) and Multi-Layer Perceptron (MLP), both achieving perfect accuracy and 0 false positives.

The majority of machine learning models, as shown in Figure 4, have an AUC of 1, which denotes infallible classification. Nevertheless, logistic regression only marginally outperforms other algorithms in class separation, with an AUC value of 0.99.

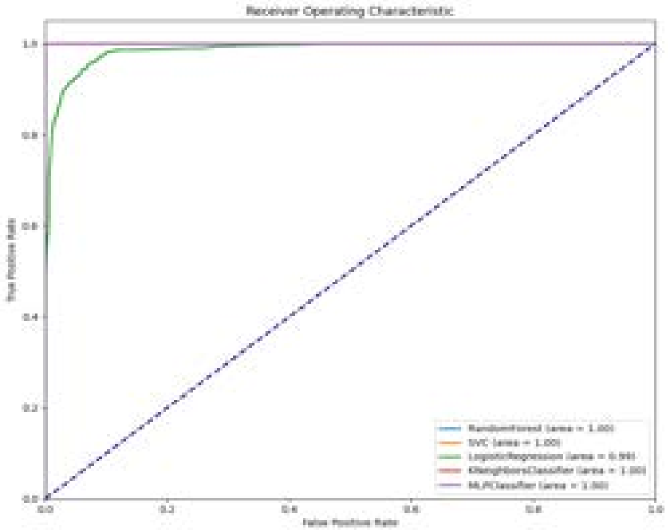


Fig. 4. ROC CURVE

### B. Deep Learning Approach

In the table V, the training performance of three deep learning models i.e., CNN, LSTM, and DNN is presented. Finally the CNN model had a high final training accuracy of 99.91% with a 10 fold CV accuracy of 99.87%. Training accuracy 99.90% with CV accuracy 99.70% was a little lower than the LSTM model. Among the rest, DNN trained with accuracy up to 99.92% and the highest CV accuracy at 99.95%.

TABLE V
DEEP LEARNING MODEL TRAINING PERFORMANCE

| **Model** | **Final Training Accuracy (%)** | **10-Fold CV Accuracy (%)** |
|---|---|---|
| CNN Model | 99.91 | 99.87 |
| LSTM Model | 99.90 | 99.70 |
| DNN Model | 99.92 | 99.95 |

Testing performance of CNN, LSTM and DNN has been presented in the table VI. The results prove the accuracy as well as the AUC scores of each model, where the model that reaches the highest accuracy is CNN that gains 99.99% and AUC score of 0.9889. Following an LSTM model that had an accuracy of 99.54% and AUC of 0.9956, the accuracy achieved by the DNN model was 99.90% and AUC of 0.9993. These results show that the performed exceptionally well with highest accuracy and perfect AUC, which means the models are effective for the current task.

It is evident from Figures 6, 5, and 7 that there is little variation in the validation accuracy and loss. Because of their stability, we may assume that these models have internalized the underlying pattern found in the data without being overfitted or underfitted. Since they show that the models can stay on course without nosediving, the little deviations shown here are encouraging for the models' predictive accuracy. Ultimately, these findings demonstrate the models' resilience in addressing the given job.

TABLE VI
DEEP LEARNING MODEL TESTING PERFORMANCE

| **Model** | **Accuracy (%)** | **AUC(%)** |
|---|---|---|
| CNN | 99.99 | 98.89 |
| LSTM | 99.54 | 99.56 |
| DNN | 99.90 | 100 |

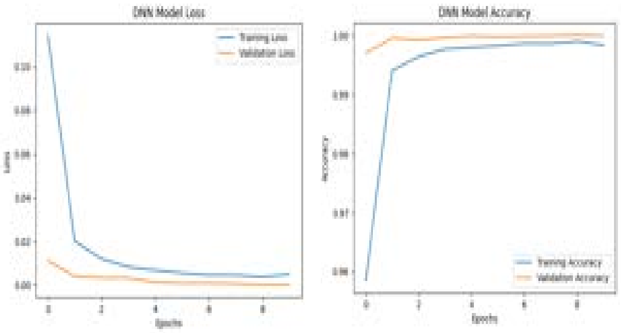


Fig. 5. Validation loss and Accuracy curve of DNN

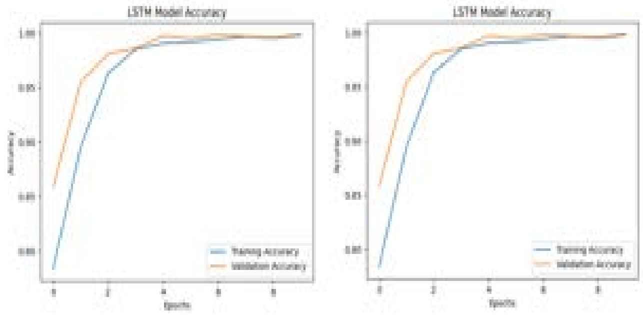


Fig. 6. Validation loss and Accuracy curve of LSTM

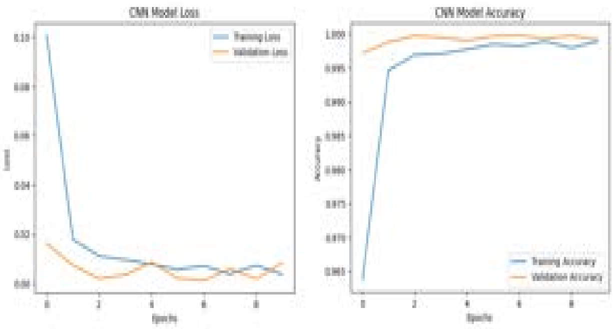


Fig. 7. Validation loss and Accuracy curve of CNN

The scores on the table VII for the deep learning models, CNN, LSTM and DNN are MCC (Matthews Correlation

Coefficient) and Kappa. It got an MCC score of 99.46% and a Kappa score of 99.46%. The Kappa score of the LSTM model achieved value 99.76% while it achieved an MCC score of 99.76%. With an MCC score of 99.99%, and Kappa of 100%, the DNN model performed well. The values of these scores suggest that these models are making reliable predictions and that the DNN model performs the best on MCC and Kappa values, which indicates that the predicted values agree very well with actual values.

TABLE VII
MCC and Kappa Score of Deep Learning Models

| Model | MCC Score (%) | Kappa Score (%) |
|---|---|---|
| CNN | 99.46 | 99.46 |
| LSTM | 99.76 | 99.76 |
| DNN | 99.99 | 100.00 |

The DNN model also has a great capacity to detect all types of malware without any omissions, as seen in Figures 8, 9 and 10, which shows the greatest accuracy in this evaluation. However, both CNN and LSTM have major drawbacks, including erroneous negative findings and, in the case of LSTM, false positive results. These findings show that in order to decrease misidentifications and increase overall detection effectiveness, these models must be further refined and implemented.

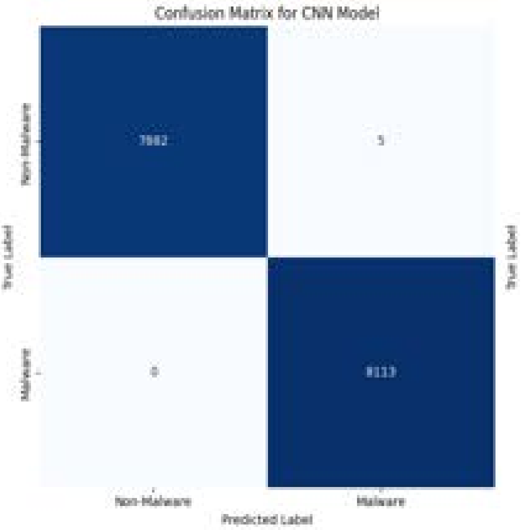


Fig. 8. Confusion Matrix of CNN

Figure 11 shown that all the deep learning model have a perfect roc curve that all curves positioned in left top corner where the roc curve is flat. This just means that all the models perfectly classify malware and not malware.

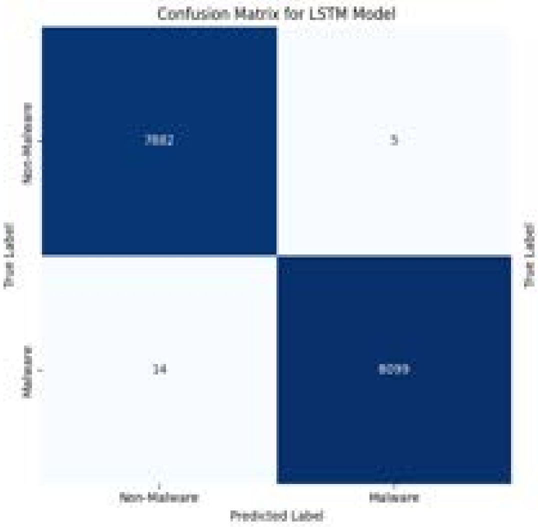


Fig. 9. Confusion Matrix of LSTM

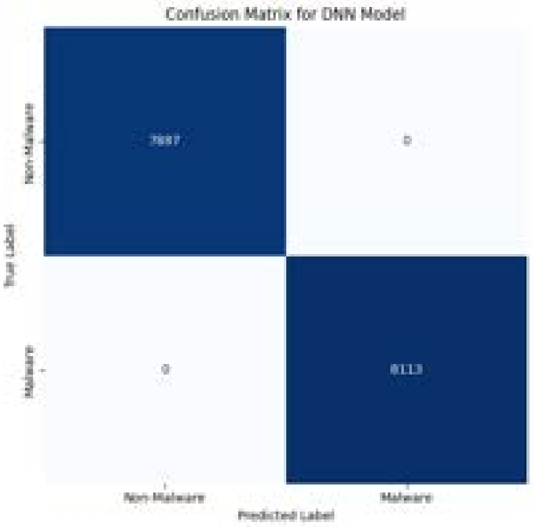


Fig. 10. Confusion Matrix of DNN

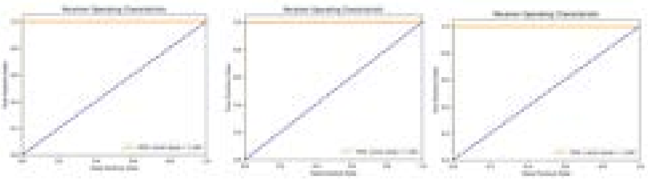


Fig. 11. ROC Curve of All Deep Learning Models

## V. Proposed Approach

The DNN model achieves top performance through 99.90% accuracy together with 0.9993 AUC and 99.99% MCC and 100% Kappa scores which indicate highly reliable predictions. The malware detection system proves to be the most suitable because it achieves stable results without producing any errors. Due to its superior performance this proposed model stands as the top selection. The hyperparameter used in DNN is given in the following table VIII.

TABLE VIII
DNN MODEL HYPERPARAMETERS

| **Hyperparameter** | **Value** |
|---|---|
| Model Type | DNN |
| Optimizer | Adam |
| Loss Function | Binary Cross-Entropy |
| Activation (Hidden) | ReLU |
| Activation (Output) | Sigmoid |
| Epochs | 10 |
| Batch Size | 32 |
| Validation Split | 0.2 |
| Dropout Rate | 0.5 |
| Hidden Layer 1 Units | 128 |
| Hidden Layer 2 Units | 64 |
| Input Shape | Based on $X_{train}.shape[1]$ |

## VI. Comparative Analysis

This current study surpasses most of the existing literature in terms of accuracy, achieving 99.99% with our proposed DNN model summarized in Table IX.

TABLE IX
COMPARISON WITH PREVIOUS STUDIES

| **Study** | **Model Used** | **Best Accuracy(%)** |
|---|---|---|
| [7] | Decision Tree | 99.00 |
| [8] | Random Forest | 99.44 |
| [9] | Deep Learning (EMBER dataset) | 87.76 |
| [10] | Random Forest | 99.24 |
| [12] | LSVC | 99.743 |
| [14] | CNN | 98.68 |
| **This Study** | DNN | 99.99 |

## VII. Conclusion

In this study we discuss how to invoke ML and DL techniques for malware detection with the aim of establishing comprehensive analysis of optimized approaches. As the level of malwares sophistication increases, traditional methods of detection are shown to be ineffective and need advanced technique. The results illustrate that the DNN is capable of identifying complex, developing threats with greater accuracy than any other model, 99.99% and a perfect AUC of 100%. Feature selection and preprocessing was highly effective for improving the model performance, which made the DNN the most reliable approach.

In the future, detecting what we call objects based on the combination of strengths of both ML and DL, and that has nothing to do with labels and classes, may improve detection accuracy even more. The performance can be further enhanced through transfer learning especially when there are only a few labeled data available. Likewise, an ability to develop real time malware detection systems, that adapt in real time to new threats through an API, is also critical. In order to develop more resilient cybersecurity, it is necessary to expand research to encompass various, rather than some datasets, and thus obtain a more complete picture of model performance.

## References

[1] D. Craft, “Malware statistics & facts: Frequency, impact & cost,” *Worth Insurance*, July 2024. [Online]. Available: https://www.worthinsurance.com/post/malware-statistics. [Accessed: Feb. 10, 2025].

[2] C. Brooks, “Cybersecurity Trends & Statistics: More Sophisticated and Persistent Threats So Far in 2023,” Forbes, May 5, 2023. [Online]. Available: https://www.forbes.com/sites/chuckbrooks/2023/05/05/cybersecurity-trends--statistics-more-sophisticated-and-persistent-threats-so-far-in-2023/. [Accessed: Feb. 10, 2025].

[3] Astra Security, “How Many Cyber Attacks Per Day? Cyber Attack Stats You Should Know,” GetAstra, Feb. 10, 2025. [Online]. Available: https://www.getastra.com/blog/security-audit/how-many-cyber-attacks-per-day/. [Accessed: Feb. 10, 2025].

[4] M. Kianpour and S. Raza, “More than malware: Unmasking the hidden risk of cybersecurity regulations,” *International Cybersecurity Law Review*, vol. 5, pp. 169–212, Feb. 2024. [Online]. Available: https://doi.org/10.1365/s43439-024-00111-7. [Accessed: Feb. 10, 2025].

[5] S. Karapoola, N. Singh, C. Rebeiro, and V. Kamakoti, “SUNDEW: A case-sensitive detection engine to counter malware diversity,” *IEEE Transactions on Dependable and Secure Computing*, pp. 1–15, 2024, doi: 10.1109/TDSC.2024.3406699.

[6] SonicWall, “Live Cyber Attack Map,” SonicWall Attack Map, 2025. [Online]. Available: https://attackmap.sonicwall.com/live-attack-map/. [Accessed: Feb. 10, 2025].

[7] Akhtar, Muhammad Shoaib, and Tao Feng. ”Malware Analysis and Detection Using Machine Learning Algorithms.” Symmetry 14, no. 11 (2022): 2304. https://www.mdpi.com/2073-8994/14/11/2304.

[8] Hussain, Abrar, Muhammad Asif, Maaz Bin Ahmad, Toqeer Mahmood, and M. Arslan Raza. ”Malware Detection Using Machine Learning Algorithms for Windows Platform.” In *Proceedings of International Conference on Information Technology and Applications*, edited by Abrar Ullah, Sajid Anwar, Álvaro Rocha, and Steve Gill, 619-632. Springer Nature Singapore, 2022.

[9] Divakarla, Usha, K Hemant Kumar Reddy, and K Chandrasekaran. ”A Novel Approach towards Windows Malware Detection System Using Deep Neural Networks.” Procedia Computer Science 215 (2022): 148-157. https://doi.org/10.1016/j.procs.2022.12.017.

[10] Azeez, Nureni Ayofe, Oluwanifise Ebunoluwa Odufuwa, Sanjay Misra, Jonathan Oluranti, and Robertas Damaševičius. ”Windows PE Malware Detection Using Ensemble Learning.” Informatics 8, no. 1 (2021): 10. https://www.mdpi.com/2227-9709/8/1/10.

[11] Akhtar, M. S., and Tao Feng. ”Evaluation of Machine Learning Algorithms for Malware Detection.” Sensors 23, no. 2 (2023): 946. https://doi.org/10.3390/s23020946.

[12] S. D. Sl and C. Jaidhar, “Windows malware detection system based on LSVC recommended hybrid features,” *Journal of Computer Virology and Hacking Techniques*, vol. 15, June 2019, doi: 10.1007/s11416-018-0327-9.

[13] M. K. H. Siam, M. Bhattacharjee, S. Mahmud, M. S. Sarkar, and M. M. Rana, “The impact of machine learning on society: An analysis of current trends and future implications,” *arXiv preprint arXiv:2404.10204*, Apr. 2024. Available: https://doi.org/10.48550/arXiv.2404.10204.

[14] Bawazeer, Omar, Tarek Helmy, and Suheer Al-hadhrami. ”Malware Detection Using Machine Learning Algorithms Based on Hardware Performance Counters: Analysis and Simulation.” Journal of Physics: Conference Series 1962 (2021): 012010. https://doi.org/10.1088/1742-6596/1962/1/012010.

[15] Amato, Alberto, and Vincenzo Di Lecce. ”Data preprocessing impact on machine learning algorithm performance.” Open Computer Science 13, no. 1 (2023): 20220278. https://doi.org/10.1515/comp-2023-0032.

[16] N. Saravana, “Malware Detection,” *Kaggle Datasets*, 2023. [Online]. Available: https://www.kaggle.com/datasets/nsaravana/malware-detection. Accessed: Feb. 10, 2025.

[17] M. E. Haque, A. Hossain, M. S. Alam, A. H. Siam, S. M. F. Rabbi, and M. M. Rahman, ”Optimizing DDoS Detection in SDNs Through Machine Learning Models,” *2024 IEEE 16th International Conference on Computational Intelligence and Communication Networks (CICN)*, 2024, pp. 426-431, doi: 10.1109/CICN63059.2024.10847458.