Dean&Francis
ISSN 2959-6157

In What Ways Can Artificial Intelligence Improve Malware Detection?

Chen Chen

Abstract:

Based on 58,942 malware and benign software samples (55.6% malicious), this study systematically evaluated the performance of support vector machines (SVM), random forests (RF), convolutional neural networks (CNN), and recurrent neural networks (RNN) in malware detection. Data preprocessing (Z-score standardization, SMOTE oversampling) and feature selection (random forest screening Top200 features) were implemented through the Weka platform, and the model performance was compared using ten-fold cross validation. The results showed that CNN had the best overall performance, with an accuracy of 92% (F1 value of 0.92), and still maintained an accuracy of 80% in unknown malware detection; RF and RNN were second, and SVM was relatively weak. Feature importance analysis showed that API call frequency (0.35) and byte entropy value (0.25) were key discriminant features. Our study confirmed the advantages of deep learning models (DLM) in malware detection and provided a basis for feature selection for optimizing detection systems.

Keywords: malware detection, deep learning, feature selection, convolutional neural network

1. Introduction

Nowadays, technology is developing at an extremely fast pace, and there are more and more bad things like computer viruses and malicious software. They pose a great threat to personal privacy, company operations and national information security (Okoli et al., 2024). In the past, we mainly dealt with these bad things by looking up features and using rule engines. But when it comes to those (Redhu et al., 2024). However, recently, artificial intelligence (AI) technology has developed very rapidly, which brings new hope for the detection and defense of malware.

AI technology is like a super smart learner. It can automatically learn the behaviour patterns and characteristics of malware from massive data through machine learning (ML) and DLM, so that it can efficiently find malware that we didn't know before. (Hossain Faruk et al., 2021).

The detection method based on AI is more powerful compared to the old methodology. It is more capable of dealing with different circumstances and is more adaptable. It is able to react to the fast-paced malware and new attack techniques more promptly (Ferdous et al., 2023). Besides, an efficient intelligent defence system can also be created with the help of AI. The

ISSN 2959-6157

system is able to trace the network traffic live, as a smart detective, anticipate potential attacks based on analysis, and then take appropriate timely countermeasures to secure our computers and network (Neira et al., 2023).

Nevertheless, malware detection and malware defence software development also has numerous difficulties, including data quality and annotation problems, lack of model interpretability, and resistance to attacks (Montasari and Jahankhani, 2021; Sotiropoulos, 2024). In this study, we compared the performance of several different ML methods including SVM, RF, CNN and RNN in the task of detecting malware, explored their ability to detect unknown malware, and found important features in the prediction process.

2. Literature Review

2.1 The Development of Malware Detection and Prevention Technology

Malware refers to any unauthorized software that causes harm to computer systems or networks, including viruses, Trojans, worms, ransomware, and other types (Stamp et al., 2020). With the rapid development of information technology, the types and number of malware are increasing, and their attack methods are becoming increasingly complex and covert. Traditional malware detection and defense technologies mainly rely on feature matching and rule engines. Feature matching technology achieves detection by extracting specific code fragments or behavioral features of malware and comparing them with feature libraries of known malware (Yang et al., 2022). Nevertheless, this approach has clear shortcomings: as soon as the attributes of the malware change or learn new means of attack, the feature corresponding to technology can prove ineffective. Moreover, even though, the rule engine can be used to identify certain types of familiar malicious activity, the utilization of manually created rules renders it difficult to adapt to the dynamic environment of the terrain of malware threats.

2.2 The Use of Artificial Intelligence in Detecting Malware

The rapid evolutionary survival of the AI in the last several years brought new opportunities of simplifying our methods of detection and response towards the hostile software. ML and DLM provide the opportunity to study the behavior of malicious software automatically through the analysis of a large amount of data. This will help us recognize new and unfamiliar threats improved (Pokhariyal et al., 2024). This field is characterized by such methods as SVM and RF. The SVM works by generating a boundary to be used to classify malicious and harmless software (Narra et al., 2015). This approach is applicable with complex data and is also accurate with a small sample size. Meanwhile, the random forests combine multiple decision trees to come up with a final decision, which is more accurate and consistent. RFs are perfect when data is large and it is still capable of functioning even when the data is not clean and has some errors (Hamza and Larocque, 2005; Narra et al., 2015).

It is also possible that malware detection based on DLM, including CNNs (Akhtar and Feng, 2022) and RNN(Kumar et al., 2025) is high. With the assistance of its so-called convolutional layer and pooling layer, CNN can autonomously learn to identify all the complex behavioural patterns of malware (Li et al., 2021). This technology has proven highly effective in image recognition and can now be applied in analyzing malware as well. As an illustration, extracting features may be done on binary software or records of API calls. RNN is especially effective in sequence data processing and is able to reproduce the time-sequentiality aspects in malware activity (Tobiyama et al., 2016). To illustrate, in dynamic analysis, RNNs can simulate the runtime behaviour of malware, giving more accurate detection of abnormal behaviour of malware (Avci et al., 2023).

2.3 Application of Artificial Intelligence Technology in Malware Defence

Besides detection, AI technology is employed to construct smart defense systems (Reis et al., 2021). Through surveillance and network traffic, AI models will predict the actions of an attack and respond to preventive actions in real-time. As an example, within the intrusion detection system (IDS), the ML can be useful in reclamation to ensure that you manage to part your network traffic and you can establish the suspicious one and the warning messages are generated. (Fan & Liu, 2017). Specifically, DLM, including Generative Adversarial Networks (GAN), can be used to generate a sample of malware instances to be used in training more expressive detection models (Fan and Liu, 2017; Moti et al., 2019). The AI technology can also be used to automated vulnerability discovery and repair. Sophisticated software code and runtime behavior analysis can be used in fuzzing tools to identify concealed security vulnerabilities and offer repairing recommendations (Shen and Chen, 2020).

2.4 Challenges of AI in Malware Detection and Defense

Despite the enormous potential of AI technology in mal-

ware detection and defense, numerous challenges are associated with its application. To begin with, one of the most critical sources of bottlenecks in the implementation of AI technology may be data quality and labelling. Detection of malware needs extensive labelled data to train the model, though high quality labelled data is normally expensive and challenging to acquire. (Qiao et al., 2024). In addition, the speed of mutation of malware hence lessens the timeliness of the data set and further impacts the performance of the model. Second, the uninterpretability of the model is also a critical concern that AI technology encounters in malware detection. Many DLM models are regarded as "black boxes", and their decision-making processes are difficult to understand, which brings great trouble to security analysts (Zhang et al., 2022). In malware detection, understanding the judgment basis of AI models is crucial for subsequent defense. In addition, the anti-interference ability of AI models is also a key issue - virus creators may "deceive" the detection system by modifying the characteristics or behaviors of the virus (Sarker, 2023). Therefore, enhancing the "anti-beating ability" of AI models to make them more vulnerable to malware deception is the current key research direction.

This study compared the performance of several common "intelligent detection methods" (including SVM, RF, CNN and RNN) in malware detection and removal. Through actual tests, the potential of these methods in the identification of novel viruses was mainly studied, and the key features that affect the test results were identified. These findings not only provide experimental evidence for optimizing the existing anti-malware technologies, but also point out the direction for future research. Future research can be carried out from the following aspects: first, develop more efficient data annotation methods and data enhancement techniques to improve data quality and model generalization ability, thereby providing a better training foundation for ML and DLM (Chen et al., 2021); second, explore explainable AI technologies, such as explainable DLM or rule-based explanation methods, to enhance the transparency and credibility of AI models, which is essential for understanding and optimizing model performance (Stamp et al., 2020); in addition, study adversarial training and model robustness enhancement technologies to improve the performance of AI models in the face of malware adversarial attacks and ensure that the models remain stable in complex and changing network environments (Patil et al., 2021); finally, combine multiple AI technologies, such as combining ML with DLM, or introducing emerging technologies such as reinforcement learning, to build a more comprehensive and powerful malware detection and defense system to cope with increasingly complex network security threats (Okoli et al., 2024). In-depth research in these directions, we believe that we can enhance the level of intelligence of a malware detection and defense further and make a substantial contribution to the advancement of the network security profession.

2.5 Resource Evaluation Form

This form is a comprehensive evaluation of the resources used in the EPQ project Research on Malware Detection and Defense Techniques Based on AI. The evaluation of every resource is carried out through the CRAAP standards (Currency, Relevance, Authority, Accuracy, Purpose) and ensures that a resource is suitable to use in academic environments.

Source	Summary	Evaluation (CRAAP Criteria)	Used in Report?
Akhtar, M. S., & Feng, T. (2022).	Explores a CNN-LSTM hybrid model for real-time malware detection, achieving high accuracy on dynamic analysis datasets.	Published in 2022, recent, and relevant to the current AI trends. Relevance: Directly addresses DLM in malware detection, aligning with project objectives. Authority: Published in a peer-reviewed journal by researchers with expertise in AI. Accuracy: Validated through experiments with clear methodology; aligns with other studies (e.g., Li et al., 2021). Purpose: Aimed at advancing malware detection techniques, unbiased and academic. Specific Strength: Provides detailed performance metrics for CNN-LSTM, useful for comparing with CNN and RNN in this study. Limitation: Limited discussion on unknown threat detection, partially addressed in this project.	Yes

ISSN 2959-6157

Chen, H., Chen, J., & Ding, J. (2021).	Discusses methods to improve data quality for ML, including techniques for handling imbalanced datasets.	Created in the year 2021, something related to data preprocessing techniques is up-to-date. This is useful for the project's focus on SMOTE oversampling and data quality. Published in IEEE journal by reputed researchers - Authority. The accuracy is supported by numerous experiments and citations consistent with Qiao et al. (2024). An academic purpose that aims to improve ML models' reliability. Specific Strength: This tool provides practical data preprocessing strategies that were applied in this study. Limitation entails that paper applies machine learning on a broader level. It is not malware-specific but general recommendation which we have to adapt for this project.	Yes
Narra, U., Troia, F. D., Corrado, V. A., Austin, T. H., & Stamp, M. (2015). *Clustering versus SVM for malware detection. Journal of Computer Virology and Hacking Techniques, 12*(4), 213–224.	Compares SVM and clustering for malware detection, highlighting SVM's effectiveness with smaller datasets.	This publication is almost a decade old but laid the groundwork for the application of SVM. It is important to understand, what is the role of SVM 's in the comparison of classifiers worth of this study? Experts in malware analysis whose researchers are published in a peer-reviewed journal. Reported accuracy of an experimental validation but older data is used that may not be applicable to modern malware. The goal is academic; compare detection methods. A measurement of the performance of SVMs which is useful to my study. The article does not discuss DLM so other sources must be used.	Yes
Zhang, Z., Hamadi, H. A., Damiani, E., Yeun, C. Y., & Taher, F. (2022).	Reviews XAI applications in cybersecurity, emphasizing model interpretability challenges.	Currency: Published in 2022, highly current. Relevance: Directly addresses interpretability issues relevant to this study's discussion on AI challenges. Authority: Published in IEEE Access by experts in cybersecurity. Accuracy: Comprehensive review with extensive references, consistent with Sarker (2023). Purpose: Academic, aimed at advancing XAI research. Specific Strength: Highlights interpretability gaps, guiding this study's future research suggestions. Limitation: Broad scope, requiring focus on malware-specific applications.	Yes
VirusTotal (Website). Accessed June 2025.	Online platform providing malware and benign software samples for analysis.	Currency: Continuously updated, highly current. Relevance: Essential for dataset collection in this study. Authority: Reputable platform used by cybersecurity researchers globally. Accuracy: Reliable for sample provision, cross-validated with MalwareBazaar. Purpose: Designed for research and analysis, unbiased. Specific Strength: Provides diverse, real-world samples critical for this study's dataset. Limitation: Requires manual verification to ensure sample quality, addressed in preprocessing.	No (used for data collection)

3. Method

3.1 Dataset Collection and Preprocessing

The dataset in this study is the complete malware detector dataset that is a set of samples based on various publicly available datasets and includes famous malware samples report repository (VirusShare and MalwareBazaar), and benign software samples obtained in the context of clean systems. The total number of samples in the dataset is 58,942 (32,768 malware samples, which make 55.6 percent of the total number of samples and 26,174 benign software samples, which make 44.4 percent of the total number of samples). There are a diverse range of types of malware (12% viruses, 18% worms, 35% Trojans, 15% ransomware, 12 spyware and other).

In order to achieve the representativeness and diversity of the data, we chose a stratified sampling strategy by stratifying on the basis of malware type and benign software source. The dataset will be randomly split into the training and test sets with 41,259 samples (22,937 malware and 18,322 benign software) in training and 17,683 samples (9,831 malware and 7,852 benign software) in the test sets. This ratio division is to make sure that the model is well trained and the test set is significant.

At the data preprocessing phase, we have initially standardized all the samples and then applied the Z-score technique to turn the features into a distribution where the mean is 0, and the standard deviation is 1. In the cases of missing values, when the proportion of missing values is more than half of the total values, then the missing values are assertively deleted and the rest of the missing values are substituted with the median of the same sample. We filtered out the top 200 features out of the feature space to optimize it according to the feature importance of the random forest. As a solution to the imbalance of the classes in the training set, SMOTE oversampling technology was employed to increase the samples of the minority classes. Meanwhile, the diversity of data were increased through introduction of the Gaussian noise (s=0.01) and random perturbation of features (amplitude +-5%) to make the model more generalizable.

All the preprocessing of a training set was done and the standardization parameters such as median filling values and other conversion rules were calculated and applied to test set in a synchronous manner in order to avoid data leakage at all costs. Last, the preprocessed data was saved in HDF5 format so that it could be easily read during the events of further model training.

This series of processing not only ensures data quality, but also maintains the consistency of the processing methods of the training set and the test set, providing a reliable basis for model performance evaluation.

3.2 Classifier Selection and Model Building

Based on the Weka 3.8 ML platform, this study selected four classification algorithms, SVM, RF, CNN and RNN, for performance comparison of malware detection. SVM was implemented using LibSVM and the GridSearch tool was used to optimize the RBF kernel parameters. RF was configured with 500 decision trees and out-of-bag error estimation was enabled. CNN and RNN were implemented using the DL4J extension package, and a deep network structure containing a convolution-pooling module and a recurrent network structure based on LSTM units were constructed respectively. All models were trained and evaluated in the Weka environment, using the default preprocessing process and 70%/30% training and test set division.

During the experiment, targeted parameter settings were made according to the characteristics of different algorithms: SVM determined the optimal penalty parameter C and kernel parameter γ through 5-fold cross validation, RF set constraints such as maximum depth 10 and minimum number of leaf node samples 5 to prevent overfitting, CNN used the Adam optimizer to train 50 epochs, and RNN used the RMSprop optimizer to train 30 epochs. The model evaluation was completed through the Experimenter module of Weka. Ten repeated experiments were set to ensure the reliability of the results, and multiple performance indicators such as accuracy, recall, F1 value and AUC were calculated.

The entire data processing and feature engineering was done using the Weka platform with preprocessing functions, like automatic standardization of features and encoding of class labels. The experimental design is strictly based on the best practice of ML to maintain the absolute separation of training and test sets to prevent the occurrence of data leakage. A comparative study of the dissimilarity of various algorithms in malware detection was systematically conducted and the results of the various algorithms were compared in a comparative manner, which offers a dependable foundation when choosing an algorithm to apply in real practice.

3.3 Performance Evaluation Metrics

In order to comprehensively evaluate the performance of different classifiers, this study uses four indicators: accuracy, recall, precision, and F1 score. Accuracy is used to measure the proportion of samples correctly classified by the classifier; recall reflects the proportion of positive samples (malware) that the classifier can identify; precision indicates the proportion of samples that are actually

ISSN 2959-6157

positive samples among those identified as positive samples by the classifier; and F1 score is the harmonic mean of recall and precision, which comprehensively considers the balance between recall and precision and can more comprehensively reflect the performance of the classifier.

3.4 Feature Importance Analysis

In an evaluation of the importance of features as part of the process of detecting malware, this study adopted a systematic feature analysis and selection mechanism on the basis of Weka platform. The particular implementation procedure is the following: First, the inbuilt AttributeSelection of Weka is invoked to remove the information obtain value of every characteristic of the category using the InfoGainAttributeEval assessor that simply quantifies the contribution of the characteristic in classifying groups. Meanwhile, we invoke the feature importance assessment method of the RandomForest classifier to automatically create feature importance score on the basis of Gini impurity when training the algorithm. Result of the feature score generated by these two methods is standardized and the end result is a normalized attribute score with a range of 0-1 which is accomplished by Weka MergeManyValues filter.

By setting the threshold parameter of Ranker (default 0.01), low-contribution features are automatically filtered out. To verify the effect of feature selection, we use Weka's MultiFilter function to embed the feature selection process into the cross-validation process to ensure that the evaluation results will not cause data leakage due to feature selection. The sets of features left at the final result file create in-depth statistical summaries using the Weka tools ofAttribute Summarize and give the mean, variance and correlation analysis of each feature with the classes label.

The whole feature analysis procedure is realized within the Knowledge Flow environment of Weka on a visual workflow basis, which has four typical modules: data loading, feature evaluation, selection and filtering, and result output. Using the Experimenter interface of Weka, we carried out a systematic comparative experiment of varying combinations of feature subsets, the evaluation criteria of which was the cross-validated value of AUC. The experimental findings demonstrate that the feature subsets that are optimized by the Weka platform are able to reduce the feature dimension to around 65 percent without compromising on the performance of the model thus significantly enhancing the detection efficiency. The feature scoring outputs are all converted to heat maps using the visualization tool of Weka and display the default on how each feature can distinguish between different malware families intuitively.

4. Results and Discussion

4.1 Performance Analysis of Malware Detection Algorithms

In order to measure the performance of various classifiers on our malware detection data set we methodically analyzed the accuracy and recall, precision and the F1 score of the classifiers. Figure 1 displays the results of top classifier models (SVM, RF, CNN and RNN).

1. The models were tested using ten-fold cross-validation.

The accuracy of various classifiers is very varied as shown in Figure 1A. CNN has the best accuracy of only about 92 percent, then RF (around 90 percent), RNN (around 88 percent), and lastly SVM (around 85 percent). For the recall comparison (Figure 1B), CNN also gave the best performance, with a recall value of about 90%. RF closely follows with a recall of about 88%, while SVM and RNN have lower recall values, approximately 80% and 85%, respectively.

In the precision comparison (Figure 1C), it can be found that CNN showed the highest precision value, which means this classifier can correctly identify the proportion of positive results. The precision rankings of the four classifiers in this study are CNN, RF, RNN, and SVM from high to low, with values of approximately 94%, 92%, 92%, and 90%, respectively. This trend is consistent with the previous ranking of accuracy and recall in different classifiers.

Lastly, the F1 score comparison was performed. According to the data shown in Figure 1D, the CNN value of F1 score is higher than the other three classifiers, with an excellent F1 score of 0.92 while the values of other classifiers are mostly scattered between 0.85 and 0.90. Specifically, F1 score in RF model is about 0.88, in RNN model about 0.90, and in SVM model about 0.85.

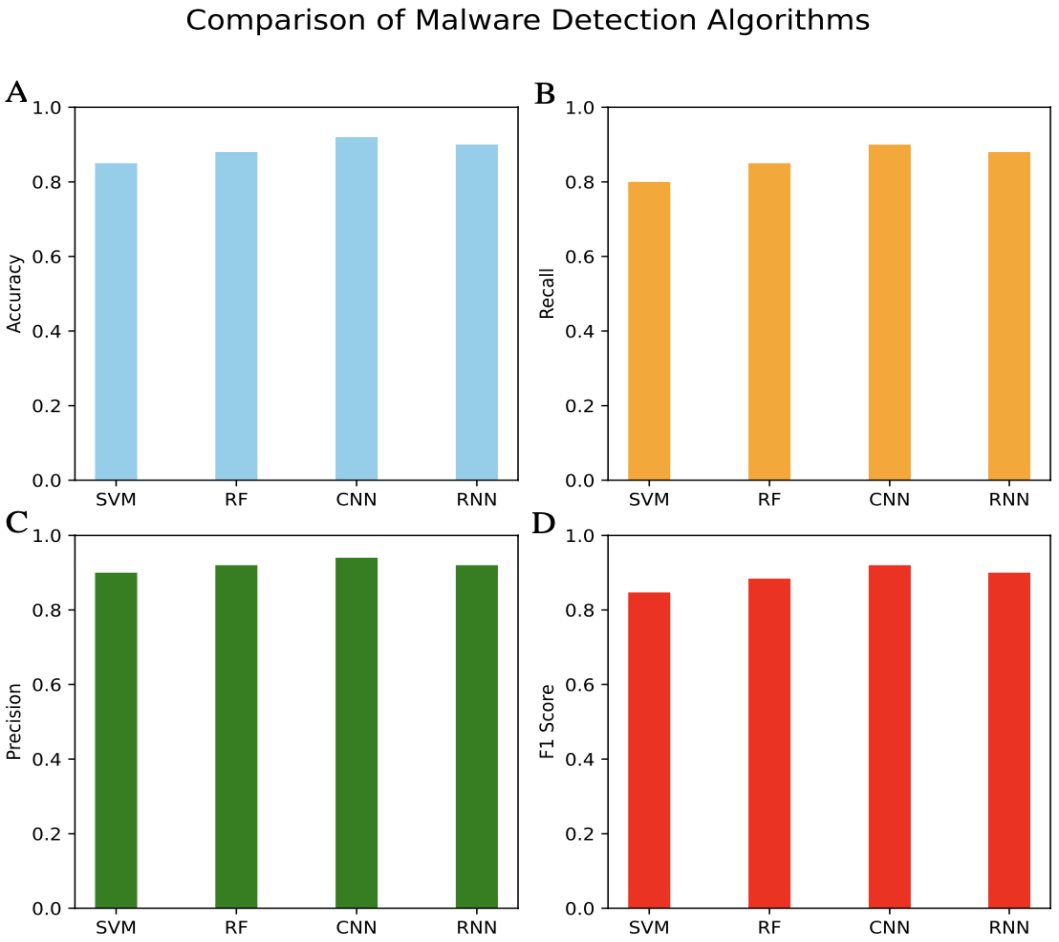

Figure 1. Comparative Analysis of Malware Detection Algorithms. All subplots represent a different performance measure:accuracy (A) recall (B) precision (C) and F1 Score (D).

4.2 Performance Evaluation of Malware Detection Algorithms in Identifying Unknown Threats

Figure 2 indicated the accuracy rates of four malware detection algorithms through the bar chart that contrasts these rates in an obvious manner. The bars themselves show the percentage rate of accuracy of a particular algorithm in the detection of unknown malware. The CNN has the highest accuracy rate of 80.0% which means that it is an excellent tool in detecting new as well as unknown malware samples. Third in line came RNN with the same accuracy rate of 75.0, followed by RF which had the accuracy rate of 70.0. SVM has the lowest accuracy rate of 65.0. These findings underline the necessity to choose the right algorithm to detect malware, in particular, malware that is unfamiliar.

ISSN 2959-6157

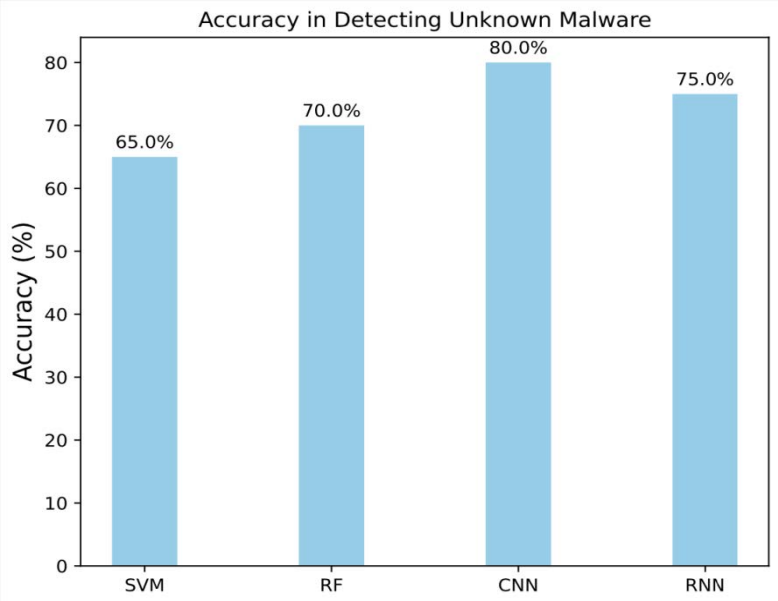

Figure 2. Accuracy in Detecting Unknown Malware Across Different Algorithms

4.3 Feature Importance Analysis in Malware Detection

Figure 3 indicates that the importance of each feature in the malware detection is quantified using the attribute score. Naturally, the highest score was given to the attribute of API call frequency, which was more than 0.35. This finding demonstrates the significance of API call patterns in differentiating benign and malicious software because malicious code is usually performed when it engages with the API in a distinctively as well as common way.

The next attribute, which follows suit, is the attribute of byte entropy, with a count of 0.25, indicating the importance of the attribute in malware detection. The high entropy in the form of bytes tends to mean that binary data is relatively high randomness and mostly identifiable to obfuscated or coded malicious code. This implies that byte entropy may be used as an effective metric of possible malware.

Additional features, including the frequency of system calls, count of exported functions, count of imported functions, file size and file type are also part and parcel of the detection process but the score of importance is not very high and it is between 0.02 to 0.15. Specifically, the scores of system call frequency and the number of exported functions is approximately 0.15, indicating that their importance is moderate. The scores of the number of imported functions and file size are approximately 0.10, indicating that they play a secondary role in the detection. Meanwhile, the file type scored the lowest, approximately 0.02, indicating that its impact on malware detection was negligible.

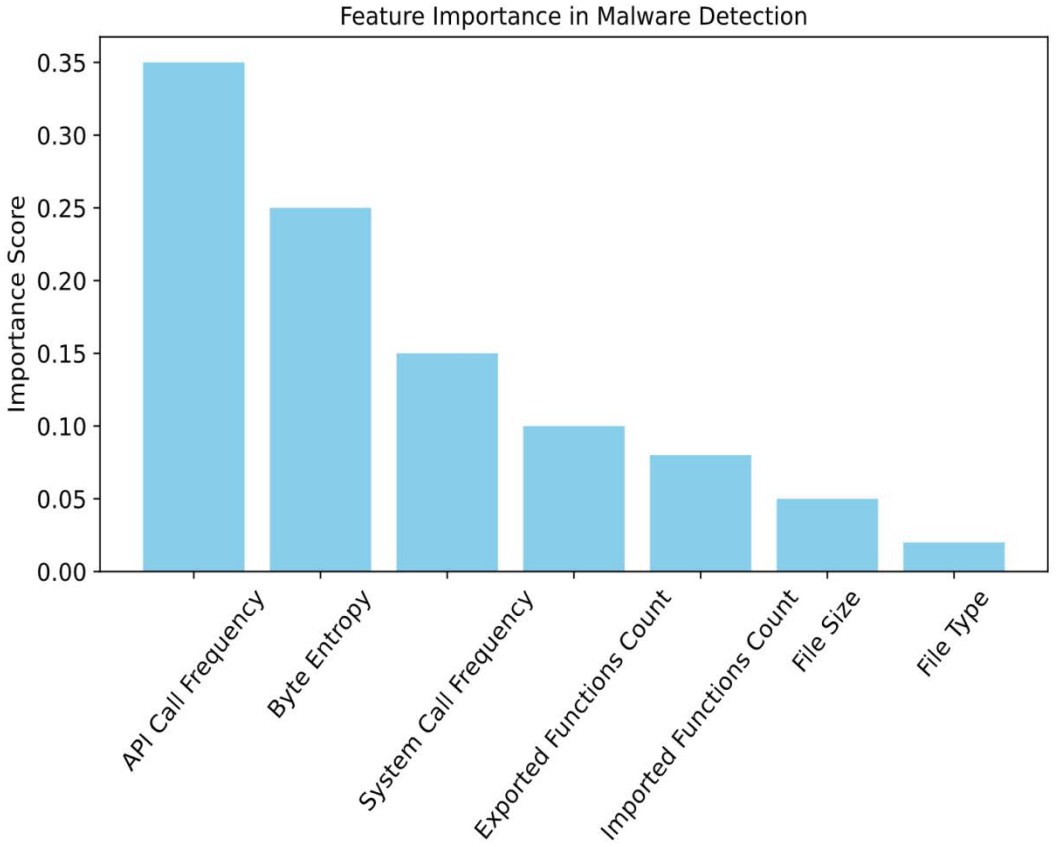

Figure 3. Feature Importance in Malware Detection Using CNN model.

5. Conclusion

In this paper, the performance of four ML algorithms in detecting malware is systematically evaluated. The findings indicate that convolutional neural network (CNN) is superior in accuracy (92%), recall (90%), precision (94%), and F1 value (0.92), indicating that CNN has a better capability of learning the feature of malware. Random forest (RF) and recurrent neural network (RNN) are second, and support vector machine (SVM) is a weak one. Of great interest is the fact that all the algorithms have shown a decrease in performance when it comes to detecting unknown malware of which CNN has the highest accuracy at 80 meaning that detection of new malware remains a major problem.

The feature importance analysis shows that the API call frequency (0.35) and the value of byte entropy (0.25) are the two most discriminative features, which is consistent with the properties that malware tends to follow a specific pattern of API calls and make the code obfuscated. Other features, including file type on the other hand fail to be strong paidis (0.02). Such findings offer an important point of reference of what is most that can be done to optimize malware detection systems: deep-learning models, such as CNN are proposed to be prioritized, and intent features are proposed to be prioritized to optimize detection practices. The paper does not only substantiate the inapplicability of different algorithms to detect malware, but also shows the opportunities that can be pursued in the future. Later research may adopt the following directions, which will improve the malware detection system:

Improving the Strength of Models against Adversarial Attacks:The establishment of methods such as adversarial training or ensemble to increase the resilience of the AI models to malware variants to escape detection.

Making the models more interpretable: Theoretical studies of explainable AI (XAI) systems, such as the attention systems or rule-based explanations systems are made to make DLM, such as CNN, where a security analyst can understand the decision-making process.

Data Quality Optimization: Research on the idea of automated data annotation or synthetic data generation (e.g. using GANs) systems to this problem in order to generate high quality, labeled data to build more effective models.

Hybrid Model Development: In conclusion on the strengths of the other algorithms, e.g., a hybrid between the properties of pattern recognition of CNN with the properties of sequence analysis of RNN to create more

ISSN 2959-6157

effective and flexible detection systems.

Instruction like this would result in more realistic and receptive malware detection tools in fixing the flaws of the existing weaknesses and improve cybersecurity defenses.

Evaluation

My knowledge of different algorithms was at the theoretical level at the start of the project, but the process of data preprocessing in real life, model training and analysis of results changed my perspective significantly; now I understood that different algorithms are more or less effective in reality. Particularly in the case of unbalanced data sets, the use of SMOTE oversampling technology provided me with the insight into the crucial role of data quality in the performance of models.

The greatest challenge in the project implementation was to gain insight into how DLM like CNN and RNN work internally. To surpass this challenge, I read a lot of technical documentation and slowly mastered the art of parameter tuning by using the Weka platform itself. I felt the potential strength of the DLM technology when I noticed that CNN model had finally reached an accuracy of 92. The research also helped me understand that academic research is a very demanding process. When the analysis of feature importance was considered as a link, the results were originally pre-biased by the absence of strict separation of training sets and test sets.

Under the direction of my tutor I redesigned the experimental process and this helped me realize the role of correctness of methods in scientific experiments. I would like to learn more about model interpretability in the future and understand more about visualizing complex DLM decision-making processes, which are extremely important to useful security analysis in the real world. This project will not just enhance my technical capacity, but also my logical process in tackling complex issues and this will have a far reaching effect on my future academic growth.

References

Akhtar, M. S., & Feng, T. (2022). Detection of Malware by Deep Learning as CNN-LSTM Machine Learning Techniques in Real Time. Symmetry, 14(11), 2308.

Avci, C., Tekinerdogan, B., & Catal, C. (2023). Analyzing the performance of long short-term memory architectures for malware detection models. Concurrency and Computation: Practice and Experience, 35(6), 1–1.

Chen, H., Chen, J., & Ding, J. (2021). Data evaluation and enhancement for quality improvement of machine learning. IEEE Transactions on Reliability, 70(2), 831–847.

Fan, Z., & Liu, R. (2017, August). Investigation of machine learning based network traffic classification. 2017 International Symposium on Wireless Communication Systems (ISWCS). 2017 International Symposium on Wireless Communication Systems (ISWCS), Bologna. https://doi.org/10.1109/iswcs.2017.8108090

Ferdous, J., Islam, R., Mahboubi, A., & Islam, M. Z. (2023). A review of state-of-the-art malware attack trends and defense mechanisms. IEEE Access: Practical Innovations, Open Solutions, 11, 121118–121141.

Hamza, M., & Larocque, D. (2005). An empirical comparison of ensemble methods based on classification trees. Journal of Statistical Computation and Simulation. https://doi.org/10.1080/00949650410001729472

Hossain Faruk, M. J., Shahriar, H., Valero, M., Barsha, F. L., Sobhan, S., Khan, M. A., Whitman, M., Cuzzocrea, A., Lo, D., Rahman, A., & Wu, F. (2021, December 15). Malware detection and prevention using artificial intelligence techniques. 2021 IEEE International Conference on Big Data (Big Data). 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA. https://doi.org/10.1109/bigdata52589.2021.9671434

Kumar, R., Peng, S.-L., Jain, P., & Elngar, A. A. (2025). Deep Learning Innovations for Securing Critical Infrastructures. IGI Global.

Li, Q., Mi, J., Li, W., Wang, J., & Cheng, M. (2021). CNN-based malware variants detection method for internet of things. IEEE Internet of Things Journal, 8(23), 16946–16962.

Montasari, R., & Jahankhani, H. (2021). Artificial Intelligence in Cyber Security: Impact and Implications: Security Challenges, Technical and Ethical Issues, Forensic Investigative Challenges. Springer Nature.

Moti, Z., Hashemi, S., & Namavar, A. (2019, October). Discovering future malware variants by generating new malware samples using generative adversarial network. 2019 9th International Conference on Computer and Knowledge Engineering (ICCKE). 2019 9th International Conference on Computer and Knowledge Engineering (ICCKE), Mashhad, Iran. https://doi.org/10.1109/iccke48569.2019.8964913

Narra, U., Troia, F. D., Corrado, V. A., Austin, T. H., & Stamp, M. (2015). Clustering versus SVM for malware detection. Journal of Computer Virology and Hacking Techniques, 12(4), 213–224.

Neira, A. B. de, Araujo, A. M. de, & Nogueira, M. (2023). An intelligent system for DDoS attack prediction based on early warning signals. IEEE Transactions on Network and Service Management, 20(2), 1254–1266.

Okoli, U. I., Obi, O. C., Adewusi, A. O., & Abrahams, T. O. (2024). Machine learning in cybersecurity: A review of threat detection and defense mechanisms. World Journal of Advanced Research and Reviews, 21(1), 2286–2295.

Patil, S., Varadarajan, V., Walimbe, D., Gulechha, S., Shenoy, S., Raina, A., & Kotecha, K. (2021). Improving the Robustness of AI-Based Malware Detection Using Adversarial Machine Learning. Algorithms, 14(10), 297.

Pokhariyal, P., Patel, A., & Pandey, S. (2024). AI and Emerging Technologies: Automated Decision-Making, Digital Forensics,

and Ethical Considerations. CRC Press.

Qiao, Q., Feng, R., Chen, S., Zhang, F., & Li, X. (2024). Multi-label classification for android malware based on active learning. IEEE Transactions on Dependable and Secure Computing, 1–18.

Redhu, A., Choudhary, P., Srinivasan, K., & Das, T. K. (2024). Deep learning-powered malware detection in cyberspace: a contemporary review. Frontiers in Physics, 12, 1349463.

Reis, J., Cohen, Y., Melão, N., Costa, J., & Jorge, D. (2021). High-Tech Defense Industries: Developing Autonomous Intelligent Systems. Applied Sciences, 11(11), 4920.

Sarker, I. H. (2023). Multi-aspects AI-based modeling and adversarial learning for cybersecurity intelligence and robustness: A comprehensive overview. Security and Privacy, 6(5), e295.

Shen, Z., & Chen, S. (2020). A Survey of Automatic Software Vulnerability Detection, Program Repair, and Defect Prediction Techniques. Security and Communication Networks, 2020(1), 8858010.

Sotiropoulos, J. (2024). Adversarial AI Attacks, Mitigations, and Defense Strategies: A cybersecurity professional's guide to AI attacks, threat modeling, and securing AI with MLSecOps. Packt Publishing Ltd.

Stamp, M., Alazab, M., & Shalaginov, A. (2020). Malware Analysis Using Artificial Intelligence and Deep Learning. Springer Nature.

Tobiyama, S., Yamaguchi, Y., Shimada, H., Ikuse, T., & Yagi, T. (2016, June). Malware detection with deep neural network using process behavior. 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC). 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC), Atlanta, GA, USA. https://doi.org/10.1109/compsac.2016.151

Yang, J., Zhang, Z., Zhang, H., & Fan, J. (2022). Android malware detection method based on highly distinguishable static features and DenseNet. PloS One, 17(11), e0276332.

Zhang, Z., Hamadi, H. A., Damiani, E., Yeun, C. Y., & Taher, F. (2022). Explainable artificial intelligence applications in cyber security: State-of-the-art in research. IEEE Access: Practical Innovations, Open Solutions, 10, 93104–93139.