AI-based Malware Threat Prediction through CNN-SVM Ensemble

Mufaro Shoniwa*, Karel Veerabudren†, Mrinal Sharma‡
School of Science and Technology
Middlesex University Mauritius
Uniciti, Flic en Flac, Mauritius
Email: *ms3232@live.mdx.ac.uk, †k.veerabudren@mdx.ac.mu, ‡m.sharma@mdx.ac.mu

***Abstract*—The dynamic terrain of malware attacks presents noteworthy obstacles to cyber security, necessitating for proactive and resilient detection techniques. Conventional signature-based methods struggle to keep pace with the new malware strains and obfuscation strategies. A systematic literature review was conducted to investigate existing methods for malware threat prediction and detection using machine learning and deep learning techniques. The review identified several promising approaches: convolutional neural networks, graph neural networks, and visual malware characterization achieving 95-99% accuracy on malware classification and detection tasks. However, major gaps were identified in the models' generalizability across diverse malware types, robustness against evasion attempts, lack of interpretability due to the black-box nature of deep learning models, and limited evaluation on real-world emerging threats as opposed to controlled datasets. This project aimed to develop an AI-based threat predictive algorithm that leverages the power of deep learning and machine learning for effective malware detection and prediction. The suggested method utilises an ensemble approach that combines a convolutional neural network (CNN) for pattern recognition in malware code structures with a support vector machine (SVM) for robust decision boundaries in the feature space, thereby enhancing generalization, interpretability and adversarial resilience. By evaluating the model on the MalImg dataset, the system achieved 92.37% accuracy. Although the developed system exhibits optimal outcomes, several areas could use more improvement. This project contributes to the ongoing efforts in combating malware threats and highlights the potential of combining deep learning and traditional machine learning techniques for effective threat prediction and detection.**

***Index Terms*—Malware, Machine Learning, Artificial Intelligence, Malware Detection, Algorithm Development**

I. Introduction

Malware, comprising worms, trojans, spyware, viruses, backdoors, and other malicious software, represents one of the most prevalent cyber threats facing governments, organizations, and individuals [1]. Research shows that in 2021, 15.45% of users worldwide experienced at least one malware attack [2]. Malware developers use this software to compromise the confidentiality, integrity, and availability of data and systems [1]. The dynamic evolution of malware has driven the development of anti-malware tools, with artificial intelligence (AI) emerging as a powerful approach [3].

Conventional signature-based malware detection techniques frequently struggle to keep pace with the speed at which new malware variants and obfuscation strategies appear [4]. Existing threat prediction models like the Common Vulnerability Scoring System use regression to quantify vulnerabilities, but lack technical assessment of human factors and provide static scores that fail to adapt as threats evolve [5]. Nearly 80% of current threat detection algorithms use individual behavior models that examine system activities, or mixed systems combining multiple approaches [6]. However, these are susceptible to being exploited as attackers adapt to the historical data and patterns they rely on.

Machine learning (ML) explores algorithms that can learn from data for prediction [7], but its reliance on manual feature extraction limits accuracy for malware recognition. Conversely, deep learning (DL) employs artificial neural networks with multiple hidden layers [7], enabling training on raw data without manual feature engineering. While deep learning has made strides in preventing cyber threats, existing ML and DL techniques still exhibit high false positive and false negative rates, necessitating hybrid approaches [4].

To address these limitations, robust algorithms use malware datasets containing real-world samples and behavioral data are needed [8]. This paper proposes an AI-based threat predictive algorithm that combines convolutional neural networks (CNNs) and support vector machines (SVMs) in an ensemble model to leverage the advantages of both DL and traditional ML for malware detection [7]. By evaluating the ensemble on the MalImg malware image dataset [8], the system aims to achieve robust malware threat prediction and detection while reducing false positives. This work contributes to continual efforts against evolving malware threats by exploring the synergistic integration of deep learning and ML algorithms.

The paper's structure is: Section II provides a background on the landscape of threat predictive algorithms and the involvement of ML and AI. Then, in Section III, works related to comparative analysis of malware detection and prediction systems are provided, before discussing the methodology used to achieve the purpose of this paper and compare the techniques used in Section IV. Section V presents the evaluation and results for the proposed solution and compared to the existing approaches. The paper is concluded in Section VI.

II. Background

Computer security has become a critical issue due to the evolution and increasing magnitude of malware activity. The

journey began in 1971 with the Creeper program, which infected ARPANET by displaying the message "I'm the creeper, catch me if you can!" [1] . Malware evolution can be divided into five phases: the early phases with basic worms and viruses, followed by the proliferation of Windows and Mail Worms, and the rise of network worms driven by the development of the Internet. From 2005 to 2016, rootkits and ransomware emerged, significantly increasing malware complexity and impact.

Currently, we are in a phase where specially crafted and weaponized malware pose severe global threats. Stuxnet exemplifies this new era, designed to sabotage Iran's nuclear program by exploiting up to four zero-day vulnerabilities in the Windows OS [9]. This evolution underscores the escalating challenges in computer security, necessitating advanced defense mechanisms to combat these sophisticated threats through predictive algorithms and AI.

A. Machine Learning and Artificial Intelligence

Artifical Intelligence aims to mimic human intelligence through techniques like Machine Learning, Natural Language Processing, computer vision and others. Machine Learning focuses on training algorithms on data to recognize patterns and make predictions. By using key steps such as data preprocessing, model training on test/train splits, evaluation of results and performance optimization, predictions on malware can be made [10].

Machine Learning encompasses supervised techniques like classification and regression, unsupervised approaches for clustering and association modelling, semi-supervised and reinforcement learning techniques. Common algorithms applied include decision trees, random forests, naive Bayes, K-nearest neighbors, support vector machines and neural networks. Additionally, deep learning models based on neural network architectures such as deep feedforward networks, convolutional neural networks and recurrent neural networks show potential for dynamic threat prediction [11].

While existing research has made advances in applying AI and ML for malware threat prediction, challenges are faced and gaps remain. Most techniques heavily rely on past attacks, making generalization to new, emerging threats difficult [12]. There is a need for predictive models that can dynamically adapt to unknown future threats. No single technique has proved perfect, suggesting an ensemble-based approach integrating multiple models may be required to holistically address the issues [12]. This motivates further research into novel architectures for accurate malware threat prediction.

B. Threat Predictive Algorithms

The landscape of malware threats has evolved significantly, progressing through various phases from basic viruses to sophisticated, weaponized malware causing major damage [13]. Notorious examples such as Stuxnet, ILOVEYOU, Code Red, Slammer, CryptoLocker and Emotet, which employed advanced techniques like AI for automating the generation of polymorphic code and machine learning for evading detection by continuously mutating their behavior patterns, have highlighted the intense and destructive nature of modern malware. This has underscored the critical need for prevention, detection and mitigation approaches that can support similar AI and ML capabilities to dynamically analyze and counter these rapidly evolving malware threats.

Traditionally, cybersecurity efforts have focused more on detecting attacks after damage has been done. However, there is an increasing realization that a shift towards predictive capabilities is necessary to forecast and mitigate threats before they can occur [14]. While subjective expert analysis has been employed, the rapid evolution of threats demands more dynamic, data-driven predictive models. Techniques like rule mining, fuzzy algorithms, decision trees, support vector machines and data mining classifiers have shown promise for malware prediction [12]. The field of Artificial Intelligence has provided powerful recognition capabilities for threat prediction with the use of ML.

III. Related Work

Studies have investigated applying machine learning and deep learning techniques for malware prediction and detection. From the systematic literature review, papers were selected based on the criteria of proposing models that achieved a minimum accuracy of 90% for malware detection and prediction.

A. Research Overview

Hemalatha et al. [15] proposed a DenseNet-based deep learning model that achieved an accuracy of 97.55% and an F1-score of 97.46% on the MalImg dataset. The approach utilized the DenseNet architecture, which supported dense connections between layers to improve feature propagation and reuse features more efficiently. DenseNets concatenate the feature maps from previous layers as inputs to subsequent layers, allowing for better flow of gradients during training and enabling the reuse of features across the network. This dense connectivity pattern helped mitigate the vanishing gradient problem and encouraged feature reuse. This made DenseNets efficient for learning discriminative representations from malware images.

Yi-Wei et al. [16] developed an intelligent multi-layered framework that classifies and analyzes malware based on the concept of modeling malware "families". The framework consists of a data preprocessing layer using autoencoders, a training layer combining machine learning algorithms (K-nearest neighbors, support vector machines, decision trees, random forests, extreme gradient boosting) with deep learning via the backpropagation algorithm, and a testing layer with activation functions to detect unknown malware.

Optimization techniques like stochastic gradient descent were also employed. This approach achieved impressive 99.98% accuracy with XGBoost and 98.88% with backpropagation on detecting unknown malware. However, the work was limited to evaluating only two datasets with no analysis of false positives or negatives. Additionally, the extensive computational requirements and lack of testing against evasion

attacks are limitations noted in other research [17].

Li et al. [18] introduced an approach utilizing graph convolutional networks (GCNs) for malware detection based on application programming interface (API) call sequences. The motivation stemmed from the diversity of malware making feature extraction challenging. By extracting the API call sequence, generating a directed cycle graph based on the Markov Chain, and designing a GCN classifier, they could effectively capture malware characteristics achieving 98.32% accuracy. A key strength was combining known malware features with test data features to improve representation. However, studies show graph-based methods can lack scalability and latency on larger datasets [19].

Cridin1 [20] presented a comprehensive implementation of a malware classification and detection system using convolutional neural networks (CNNs). This system converts malware binaries into grayscale images, allowing the CNNs to effectively learn and identify distinctive patterns associated with various types of malware. The architecture of the CNNs used includes multiple convolutional layers, pooling layers, and fully connected layers, optimized through techniques such as dropout and batch normalization to enhance generalization and reduce overfitting. The implementation achieved notable accuracy (94.67%) in distinguishing between malware and benign samples, showcasing the potential of deep learning in the cybersecurity domain. However, the evaluation of this system was primarily based on the MalImg dataset, with limited exploration of other datasets or real-world scenarios. Additionally, it does not extensively address issues such as model interpretability or robustness against adversarial attacks, which are critical for practical deployment.

Mallik et al. [21] , "Conrec: Malware Classification Using Convolutional Recurrence," presents a novel malware classification method combining convolutional neural networks (CNNs) and recurrent neural networks (RNNs). This hybrid approach utilizes CNNs to extract spatial features from malware images and RNNs to capture temporal dependencies, achieving a notable classification accuracy of 93.9% across various malware families. Despite its impressive performance, the study's evaluation was limited to specific datasets and did not fully address computational efficiency. Nonetheless, Conrec significantly advances malware detection by integrating spatial and sequential analysis, offering a robust framework for identifying complex malware patterns.

Han et al. [22] proposed MalInsight, a multi-perspective profiling framework extracting features across basic structures, low-level behaviors (APIs, DLLs), and high-level behaviors (file, registry, network operations). This systematic profiling enabled capturing a comprehensive feature space. Applying machine learning algorithms like decision trees and XGBoost, MalInsight achieved 99.76% detection accuracy and 94.2% family classification accuracy. The quantitative analysis provided insights into the importance of different profile views. A limitation is the framework's potential vulnerability to malware obfuscating monitoring logs used for behavior profiling.

B. Summary

The body of research demonstrates impressive malware detection accuracy using advanced AI/ML techniques, common limitations persist. These include the reliance on simulated datasets rather than real-world dynamic malware [23], vulnerability to adversarial evasion attacks [24], lack of interpretability in deep learning models [25], and unclear feasibility for practical deployment against evolving threats [26]. An ensemble system integrating the strengths of multiple approaches may help bridge these gaps in generalization, robustness, and transparency. By incorporating a broader spectrum of data inputs and scenarios, an ensemble model can improve generalization. The ensemble approach can strengthen defenses against adversarial attacks by creating multiple detection layers, thereby complicating evasion attempts. To tackle the interpretability, the system combines straightforward models with more sophisticated ones, striking a balance between complexity and explainability. Recent studies suggest enhancing adversarial robustness by adopting adversarial training [27] and employing defensive distillation techniques [28] , which have shown promise in improving model resilience against sophisticated evasion tactics.

IV. Proposed Solution

To address the widespread threat of malware attacks, an ensemble model that combines convolutional neural networks (CNNs) and support vector machines (SVMs) for malware prediction and detection is proposed. This approach leverages the pattern recognition abilities of CNNs and the feature engineering capabilities of SVMs. The model utilizes the MalImg: Malware Image Dataset [29], containing a diverse range of malware samples in grayscale image format. This dataset undergoes thorough pre-processing steps such as data augmentation, resizing to 28x28 pixels, converting to tensors, and normalization. The quality of the dataset is evaluated both before and after pre-processing to confirm its appropriateness. This dataset is favorable for building an image classification model for effective malware detection, as demonstrated by prior studies [30].

The CNN component of the ensemble analyzes the raw visual patterns in the malware images, leveraging the proven effectiveness of CNNs in recognizing discriminative textures and gradients for classification tasks [31]. The SVM component classifies malware based on feature vectors extracted from the images, utilizing the ability of SVMs to find optimal decision boundaries suitable for large malware domains [32]. Systematic testing is performed on different configurations of both individual models and their ensemble. Hyperparameters for the CNN (dropout rates, optimizers, learning rates) and SVM (kernels, class weights, regularization) are fine-tuned. The ensemble is evaluated with combination strategies and weight distributions. Performance is measured using accuracy, precision, recall, and F1-score metrics, offering a thorough evaluation.

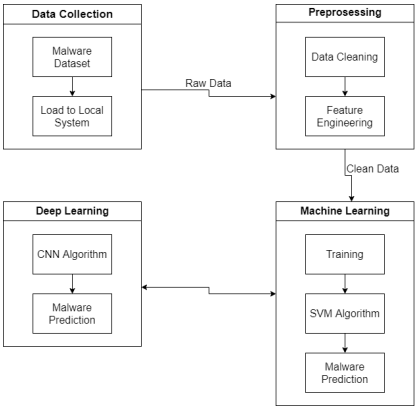

Fig. 1. Ensemble model architecture.

A. Implementation Components

1) Data Preprocessing: This module loads the malware image dataset, applies data augmentation techniques (random horizontal/vertical flips and rotations), resizes images to a consistent 28x28 pixel resolution, converts them to tensors, and normalizes the data. It splits the dataset into training and testing sets.

2) CNN Model: Using PyTorch, this module includes convolutional layers with 16 and 32 filters of 3x3 size, ReLU activations, max-pooling operations, and fully connected layers. The CNN is trained to learn visual patterns from the raw malware image data.

3) SVM Model: Utilizing the scikit-learn library, this module implements a Support Vector Classification model with a linear kernel and balanced class weights to mitigate class imbalance [33]. The SVM is trained on feature vectors extracted from the malware images.

4) Evaluation and Visualization: Ensemble and Evaluation: This module integrates the predictions from the CNN and SVM models, using a combination of logical operations to determine the final prediction based on the agreement or disagreement between the two models. It calculates accuracy, precision, recall, and F1-score at the family level. This module generates a bar graph to visualize the distribution of predictions across the malware families.

5) Main Model: The main script orchestrates the entire process, loading data, training the models, making predictions, and evaluating the performance.

Algorithm 1 Data Loading and Processing

Function load_data(data_dir):
  Define image transformations; Create dataset and get malware families
  Split data into training and testing sets; Create data loaders
  return train_loader, test_loader, malware_families
Function find_image_files(directory):
  Find image files recursively; **return** list of image file paths
Function get_malware_families(data_dir):
  Get list of malware families from directory; **return** list of family names =0

Algorithm 2 CNN Model

Class CNNModel:
  Method *init*(num_classes):
    Initialize layers
  Method *forward*(x): Perform forward pass
  Method *train*(X_train, y_train, X_val, y_val, criterion, optimizer, epochs=10):
    Train the model
  Method *predict*(X_test): Make predictions =0

Algorithm 3 SVM Model

Class SVMModel:
  Method *init*(): Initialize SVM classifier
  Method *train*($X_{\text{train}}, y_{\text{train}}$): Train the SVM model
  Method *predict*(X_{test}): Predict labels for test data
  Method *evaluate*($X_{\text{val}}, y_{\text{val}}$): Evaluate the SVM model on validation data =0

Algorithm 4 Malware Threat Predictor

Import necessary libraries
Define main function
  Load data from directory
  Split training data into training and validation sets
  Initialize models (CNN and SVM)
  Train the CNN model
  Preprocess and predict using CNN and SVM models
  Evaluate predictions
  Print evaluation results; Plot predictions
end =0

V. Evaluation and Results

The malware prediction system was evaluated using an experimental approach on the MalImg malware image dataset. The dataset was divided into stratified training (80%) and test (20%) sets to maintain even distribution of different malware families and types found in the overall data. This stratification ensured the training set represented the family/type proportions in the complete dataset, while the test set mirrored these distributions to enable unbiased evaluation of generalization capabilities across the diverse malware landscape. The models' effectiveness was assessed using accuracy, precision, recall, and F1-score metrics. Family-level evaluation examined how well the system categorized samples into their respective malware families. Different configurations of the CNN and SVM models were systematically evaluated through the following experiments:

- CNN Experiments: Varying dropout rates, optimization algorithms, and learning rates
- SVM Experiments: Testing different kernel functions, class weight techniques for imbalanced data, and regularization parameters
- Ensemble Experiments: Varying the ensemble strategy (majority voting or weighted averaging) and weight contributions of the CNN and SVM models

A. Results and Analysis

TABLE I
CNN MODEL RESULTS

Exp.	Conv. Layers	Kernel Size	Dropout Rate	Optimizer (LR)	Val. Acc. (%)	Test Acc. (%)
Baseline	Conv1 (3, 16), Conv2 (16, 32)	3x3	0.25	Adam (0.001)	90.83	86.68
Exp. 1	Conv1 (3, 16), Conv2 (16, 32)	3x3	0.25	Adam (0.0005)	89.60	73.38
Exp. 2	Conv1 (3, 16), Conv2 (16, 32)	3x3	0.3	Adam (0.001)	92.00	79.46
Exp. 3	Conv1 (3, 16), Conv2 (16, 32)	3x3	0.4	Adam (0.001)	91.77	76.73
Exp. 4	Conv1 (3, 16), Conv2 (16, 32)	3x3	0.4	Adam (0.001)	91.77	76.73
Exp. 5	Conv1 (3, 16), Conv2 (16, 32)	3x3	0.2	SGD (0.01)	60.79	71.39
Exp. 6	Conv1 (3, 16), Conv2 (16, 32)	3x3	0.3	SGD (0.01)	60.54	67.71
Exp. 7	Conv1 (3, 16), Conv2 (16, 32)	3x3	0.4	SGD (0.005)	48.64	52.70
Exp. 8	Conv1 (3, 16), Conv2 (16, 32)	3x3	0.2	RMSProp (0.001)	92.10	78.99
Exp. 9	Conv1 (3, 16), Conv2 (16, 32)	3x3	0.3	RMSprop (0.0005)	81.40	72.25
Exp. 10	Conv1 (3, 16), Conv2 (16, 32)	3x3	0.4	RMSprop (0.001)	91.99	78.71

1) CNN Experiments: The baseline CNN configuration with Conv1(3,16), Conv2(16,32) filters, 3x3 kernel, stride 1, 0.25 dropout, and Adam optimizer (LR=0.001) achieved 90.83% validation and 86.68% test accuracy (Table I). Increasing the dropout rate to 0.3 or using RMSprop slightly improved validation accuracy but not test accuracy.

TABLE II
SVM EXPERIMENT RESULTS

Exp.	Kernel	Class Weight	C (Reg.)	Val. Acc.	Test Acc.
Baseline	linear	balanced	(Default)	90.10	81.42
Exp. 1	rbf	balanced	(Default)	82.81	73.35
Exp. 2	poly	balanced	(Default)	73.44	61.89
Exp. 3	linear	None	(Default)	93.24	85.81
Exp. 4	linear	balanced	0.1	87.63	76.25
Exp. 5	linear	balanced	1.0	90.97	83.16
Exp. 6	linear	balanced	10.0	90.90	88.37
Exp. 7	rbf	None	(Default)	90.10	69.40
Exp. 8	rbf	balanced	0.1	43.55	40.00
Exp. 9	rbf	balanced	1.0	83.14	61.94
Exp. 10	rbf	balanced	10.0	90.57	67.57
Exp. 11	poly	None	(Default)	82.81	59.11
Exp. 12	poly	balanced	0.1	69.77	51.65
Exp. 13	poly	balanced	1.0	56.45	23.14
Exp. 14	poly	balanced	10.0	87.49	68.25

2) SVM Model Results: The linear kernel with balanced class weights and C=10.0 achieved the highest 88.37% test accuracy (Table II). The RBF and polynomial kernels performed worse than the linear kernel on this dataset.

TABLE III
ENSEMBLE EXPERIMENT RESULTS

Exp.	Ensemble Strategy	CNN Weight	SVM Weight	Acc.	Precision	Recall	F1-Score
Baseline	Majority Voting	0.5	0.5	88.56	86.49	77.63	77.25
Exp. 1	Weighted Avg.	0.6	0.4	81.96	75.03	77.13	73.82
Exp. 2	Weighted Avg.	0.7	0.3	88.08	75.38	76.51	74.00
Exp. 3	Majority Voting	0.6	0.4	77.45	82.76	72.06	71.57
Exp. 4	Weighted Avg.	0.8	0.2	89.26	81.30	79.27	77.09
Exp. 5	Weighted Avg.	0.9	0.1	92.37	86.42	84.83	84.59
Exp. 6	Majority Voting	0.7	0.3	88.18	86.69	80.78	80.22
Exp. 7	Weighted Avg.	0.4	0.6	79.69	72.41	71.96	69.14
Exp. 8	Majority Voting	0.8	0.2	86.30	86.86	78.57	79.01
Exp. 9	Weighted Avg.	0.3	0.7	79.54	75.88	72.56	72.00
Exp. 10	Majority Voting	0.9	0.1	80.75	81.33	74.90	73.92

3) Ensemble Model Results: The weighted average ensemble with 0.9 weight for CNN and 0.1 for SVM achieved the best 92.37% accuracy and balanced 84.83% F1-score (Table III). Majority voting with equal weights had high but slightly lower metrics.

TABLE IV
COMPARISON OF APPROACHES

Approach	Accuracy	Precision	Recall	F1-Score
DenseNet-Based DL Model [15]	97.55%	97.43%	97.50%	97.46%
XGBoost ML Algorithm [16]	99.98%	99.94%	99.94%	-%
GCN Algorithm [18]	94.67%	94.94%	93.21%	95.05%
CNN Algorithm [20]	94.67%	94.94%	93.21%	95.05%
Conv. Recurrence Algorithm [21]	93.92%	98.90%	76.02%	85.98%
Developed CNN-SVM Ensemble	**92.37%**	**84.42%**	**84.69%**	**84.83%**

4) Comparison to prior work: As summarized in Table IV, the developed CNN-SVM ensemble achieved competitive 92.37% accuracy and 84.83% F1-score on the MalImg dataset compared to methods. The discrepancies in the performance metrics can be attributed to the characteristics of the datasets used compared to the MalImg dataset, the model architecture differences and the evaluation methods. The ensemble method's class imbalance handling has resulted in balanced precision and recall scores despite the overall percentage appearing lower. Such factors show the complexity of malware detection model comparison across each of the studies. The CNN-SVM ensemble shows promising results by combining deep representation learning and robust decision boundaries.

B. Findings

Key challenges faced during development included selecting a representative and manageable dataset and working with malware images that caused preprocessing issues. The MalImg dataset has limitations in quantity and lacks coverage of the latest malware strains compared to datasets used in production environments [33]. Working with the image representations of malware binaries presented challenges in ensuring compatibility with the model architectures. The model's image-based, therefore requires malware format conversions. The model lacks real-time monitoring capabilities as there is no network connection due to the chosen dataset and limited resources [34]. The model may not adapt well to novel malware types and contains black-box elements that hinder decision exploration.

Based on these limitations and challenges, future work could expand the approach to incorporate multiple modalities such as static features and visual patterns for a comprehensive model [35]. Developing strategies for continuous learning and adaptation to emerging malware trends could enable the system to stay ahead of the evolving threat landscape. Investigating the applicability of the CNN-SVM ensemble approach to other cybersecurity domains like network intrusion detection or offline analysis could broaden its impacT. Creating a file-to-image converter component could enable range expansion of analyzable file types.

VI. CONCLUSION AND FUTURE WORK

This study aimed to create a reliable and accurate malware prediction system by using an ensemble technique combining convolutional neural networks (CNNs) and support vector machines (SVMs). The CNN-SVM ensemble model was implemented and evaluated on the MalImg malware image dataset, achieving competitive performance with 92.37% accuracy, 84.42% precision, 84.69% recall, and 84.59% F1-score.

The model demonstrated strong resilience to class imbalances; a major challenge in malware classification tasks, by utilizing the robust decision boundary construction of SVMs and the visual pattern recognition capabilities of CNNs. This emphasizes how deep learning for discriminative representation learning and classical machine learning techniques for robust decision surfaces can be combined to enhance malware detection [31]. Although the study shows a proof-of-concept, the theoretical nature of the approach is evident in its lack of real-world applicability. This conceptual groundwork illustrates a further gap for real-word testing and evaluation of current threats to bridge the gap between such models and deployable solutions.

By addressing limitations, challenges and pursuing suggested future research directions, the developed malware prediction system could potentially be further improved, optimized, and deployed as a useful tool in the ongoing fight against constantly changing malware threats. This would better meet the goals of offering a reliable, accurate, and flexible solution for malware detection and prediction.

REFERENCES

[1] T. Rains, "Cybersecurity Threats, Malware Trends, and Strategies: Learn to mitigate exploits, malware, phishing, and other social engineering attacks," Packt Publishing Ltd., 2020.

[2] D. Farhat and M.S. Awan, "A brief survey on ransomware with the perspective of internet security threat reports," in 2021 9th International Symposium on Digital Forensics and Security (ISDFS), June 2021, pp. 1-6.

[3] C. Zhang and Y. Lu, "Study on artificial intelligence: The state of the art and future prospects," Journal of Industrial Information Integration, vol. 23, p. 100224, 2021.

[4] N. Idika and A.P. Mathur, "A survey of malware detection techniques," Purdue University, vol. 48, no. 2, pp. 32-46, 2007.

[5] V. Jaganathan, P. Cherurveettil, and P.M. Sivashanmugam, "Using a prediction model to manage cyber security threats," The Scientific World Journal, 2015.

[6] I.A. Gheyas and A.E. Abdallah, "Detection and prediction of insider threats to cyber security: a systematic literature review and meta-analysis," Big Data Analytics, vol. 1, no. 1, p. 6, 2016.

[7] P. Ongsulee, "Artificial intelligence, machine learning and deep learning," in 2017 15th International Conference on ICT and Knowledge Engineering (ICT&KE), Nov. 2017, pp. 1-6. IEEE.

[8] D. Gavriluţ, M. Cimpoeşu, D. Anton, and L. Ciortuz, "Malware detection using machine learning," in 2009 International Multiconference on Computer Science and Information Technology, Oct. 2009, pp. 735-741. IEEE.

[9] A. Thabet, "Stuxnet malware analysis paper," in Code Project, 2011.

[10] R.Y. Choi, A.S. Coyner, J. Kalpathy-Cramer, M.F. Chiang, and J.P. Campbell, "Introduction to machine learning, neural networks, and deep learning," Translational Vision Science & Technology, vol. 9, no. 2, pp. 14-14, 2020.

[11] S. Yuan and X. Wu, "Deep learning for insider threat detection: Review, challenges and opportunities," Computers & Security, vol. 104, p. 102221, 2021.

[12] M.N.R. Mahrin, S. Chuprat, A. Subbarao, A.F.M. Ariffin, M.Z.A. Talib, M.Z.A. Darus, and F.A.A. Aziz, "Malware prediction algorithm: Systematic review," Journal of Theoretical & Applied Information Technology, vol. 96, no. 14, 2018.

[13] M.N. Alenezi, H. Alabdulrazzaq, A.A. Alshaher, and M.M. Alkharang, "Evolution of malware threats and techniques: A review," International Journal of Communication Networks and Information Security, vol. 12, no. 3, pp. 326-337, 2020.

[14] M. Husák, J. Komárková, E. Bou-Harb, and P. Čeleda, "Survey of attack projection, prediction, and forecasting in cyber security," IEEE Communications Surveys & Tutorials, vol. 21, no. 1, pp. 640-660, 2018.

[15] J. Hemalatha, S. A. Roseline, S. Geetha, S. Kadry, and R. Damaševičius, "An efficient densenet-based deep learning model for malware detection," Entropy, vol. 23, no. 3, p. 344, 2021.

[16] Y.W. Ma, J.L. Chen, W.H. Kuo, and Y.C. Chen, "AI@nti-Malware: An intelligent framework for defending against malware attacks," Journal of Information Security and Applications, vol. 65, p. 103092, 2022.

[17] D. Gibert, C. Mateu, and J. Planes, "The rise of machine learning for detection and classification of malware: Research developments, trends and challenges," Journal of Network and Computer Applications, vol. 153, p. 102526, 2020.

[18] S. Li, Q. Zhou, R. Zhou, and Q. Lv, "Intelligent malware detection based on graph convolutional network," The Journal of Supercomputing, vol. 78, no. 3, pp. 4182-4198, 2022.

[19] B. Kolosnjaji, A. Zarras, G. Webster, and C. Eckert, "Deep learning for classification of malware system call sequences," in AI 2016: Advances in Artificial Intelligence: 29th Australasian Joint Conference, Hobart, TAS, Australia, December 5-8, 2016, Proceedings, vol. 29, pp. 137-149, Springer International Publishing, 2016.

[20] cridin1, "Cridin1/malware-classification-CNN: This github repository contains an implementation of a malware classification/detection system using convolutional neural networks (cnns)," GitHub.

[21] A. Mallik, A. Khetarpal, and S. Kumar, "ConRec: malware classification using convolutional recurrence," Journal of Computer Virology and Hacking Techniques, vol. 18, no. 4, pp. 297-313, 2022.

[22] W. Han, J. Xue, Y. Wang, Z. Liu, and Z. Kong, "MalInsight: A systematic profiling based malware detection framework," Journal of Network and Computer Applications, vol. 125, pp. 236-250, 2019.

[23] K. Allix et al., "A Forensic Analysis of Android Malware–How is Malware Written and How it Could Be Detected?," in Proc. 2014 IEEE 38th Annual Computer Software and Applications Conference, Jul. 2014, pp. 384-393.

[24] M. Shahpasand, L. Hamey, D. Vatsalan, and M. Xue, "Adversarial attacks on mobile malware detection," in Proc. 2019 IEEE 1st International Workshop on Artificial Intelligence for Mobile (AI4Mobile), Feb. 2019, pp. 17-20.

[25] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, and S. Venkatraman, "Robust intelligent malware detection using deep learning," IEEE Access, vol. 7, pp. 46717-46738, 2019.

[26] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," in Proc. 2018 ACM SIGSAC Conference on Computer and Communications Security, Oct. 2018, pp. 2154-2156.

[27] A. Lamb, V. Verma, J. Kannala, and Y. Bengio, "Interpolated Adversarial Training: Achieving Robust Neural Networks Without Sacrificing Too Much Accuracy," in Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, 2019, pp. 95-103.

[28] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks," in 2016 IEEE Symposium on Security and Privacy (SP), IEEE, 2016, pp. 582-597.

[29] J. Moon, S. Kim, J. Song, and K. Kim, "Study on Machine Learning Techniques for Malware Classification and Detection," KSII Transactions on Internet & Information Systems, vol. 15, no. 12, 2021.

[30] G. P. Zhang, "Neural networks for classification: a survey," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 30, no. 4, pp. 451-462, 2000.

[31] N. Stakhanova, S. Basu, and J. Wong, "A taxonomy of intrusion response systems," International Journal of Information and Computer Security, vol. 1, no. 1-2, pp. 169-184, 2007.

[32] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," Journal of Artificial Intelligence Research, vol. 16, pp. 321-357, 2002.

[33] R. Ali, A. Ali, F. Iqbal, M. Hussain, and F. Ullah, "Deep learning methods for malware and intrusion detection: A systematic literature review," Security and Communication Networks, 2022.

[34] K. Allix, T. F. Bissyandé, J. Klein, and Y. Le Traon, "Are your training datasets yet relevant? an investigation into the importance of timeline in machine learning-based malware detection," in International Symposium on Engineering Secure Software and Systems, Mar. 2015, pp. 51-67, Cham: Springer International Publishing.

[35] N. Udayakumar, V. J. Saglani, A. V. Cupta, and T. Subbulakshmi, "Malware classification using machine learning algorithms," in 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI), May 2018, pp. 1-9. IEEE.