Journal of Information Systems Engineering and Management
2025, 10(36s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Advanced Framework for Detecting Malware in Portable Executable (PE) Files Using a Multi-Model

Aula Hamed Naji Al-ojaimi1
[1]*University of Thiqar, Iraq*
aula.hamed@utq.edu.iq

ARTICLE INFO

Received: 18 Dec 2024

Revised: 10 Feb 2025

Accepted: 28 Feb 2025

ABSTRACT

This paper presents Malware detection in Portable Executable (PE) files remains a critical challenge in cybersecurity, with attackers increasingly using obfuscation, polymorphism, and zero-day exploits to evade detection. Malware has emerged as a major problem in today's digital era. The malware goals are to interfere with, damage, or compromise information system and computer system without the operator's approval or knowledge. At present, malware is considered among the most common cyber threat We combine static, dynamic, and hybrid analysis techniques with ensemble learning to achieve superior detection accuracy compared to traditional methods. Our system integrates feature engineering from PE headers, byte-level CNN analysis, API sequence modeling with LSTMs, and attention mechanisms via Transformers, culminating in a stacked ensemble classifier. Experimental results demonstrate 98.7% detection accuracy with only 0.8% false positives on a dataset of 100,000 PE files. The paper provides complete mathematical formulations, architectural details, and empirical validation of our approach.

Keywords: Portable Executable (PE) Malware Detection, Multi-Model Learning, Ensemble Methods, Explainable AI, Windows Security.

I. INTRODUCTION

Preservation information systems and computer networks against malware has emerged as one of the most formidable and intricate challenges in the empire of cyber. With the propagation of the Internet, the results of malware have increased to a risky level that cannot be ignored. Reprobates exploit malware to penetrate computer systems, appropriate sensitive information, and impose important pecuniary damage. Given that the Windows operating system is the most extensively exploited stage generally, it have improve a prime aim for wicked software.. Av tests Organization has described the occurrence of recently identified malware, with over 85% engineered to target the Operating of windows system, in blatant contrast to operation systems. Additionally, closely Ninety percent of these malicious programs are definitely aimed at executable files, rather than collection. The kind of data employed for executable, dlls, purpose code, else data within the using of windows systems are denoted to as the PE data type. appropriate format to together 64 & 32-bit kinds of Windows. The data structure of PE file provides the essential data for the operation system (OS) intend to efficiently achieve the summarized practicable code. A PE file injector constitutes a specific category of malware that distributes by participating or summarizing malicious code within another Portable Executable (PE) files on a conceded system. Malware sensing skills can be primarily categorised into dynamic and static approaches. Detection requires the assessment of code manner versus a illustrious source malicious initials wanting implementing the code, in that way simplifying efficient identification of established malware. On the other hand, this method is forced in its ability to detect zero-day malware and polymorphic variants, and it struggles with challenges posed by the exponential growth of the initials database, which adversely impacts matching time. Conversely, dynamic malware analysis necessitates the execution of malware code within a controlled situation to inspect its interactions with the system. Although this method is computationally intensive, dynamic analysis fails to encompass all possible execution routes. Recently, there have been a distinct emphasis on leveraging machine learning for malware detection, mainly through the

Journal of Information Systems Engineering and Management
2025, 10(36s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

application of managed learning methods. Algorithms of Machine learning activity the efficiency and rapidity of static analysis of malware while demanding fewer resources calculations related to dynamic analysis.

1.1 Problem Significance

Portable Executable (PE) files keep on the primary vector for Windows malware, with over 1.2 million new models detected monthly (AV TEST, 2023). Traditional signature-depend on detection fails against:

- Polymorphic malware (mutation rates >90%)

- Fileless attacks (50% increase since 2021)

- Zero-day exploits (median detection time: 7 days)

1.2 Suggested Solution

Our multi-model system reports these challenges through:

1. Heterogeneous Feature Extraction*

  - Static: PE headers, entropy, import tables

  - Dynamic: API call graphs, registry changes

  - Hybrid: Opcode sequences, memory artifacts

2. Diverse Model Architecture

f(x) = g(\text{CNN}(x_{\text{bytes}}), \text{LSTM}(x_{\text{API}}}), \text{Transformer}(x_{\text{opcodes}}}))

where g is a meta-learner

3. Explainable Detection

  - SHAP values for feature importance

  - Attention visualization in API sequences

2. BACKGROUND

The operation of deep learning procedures for the Portable Executable (PE) malware detection is an array of methods have been expressed in the existing literature. [3] meticulously tested various approaches for feature extraction engaged in previous machine learning based malware detection systems, while also assessing the classification methods utilized. Their primary emphasis was on the precision rates associated with each of the formerly established classifiers. They decided that a deep learning example could potentially harvest an advanced anti-malware resolution. [4] provided a comprehensive review of malware detection practices that influence deep acquisition. They posited that the efficacy of a sensing system of malware is deeply depending upon the features take out and the classification or gathering approaches employed. For the feature extraction procedure, they supported for the implementation of static analysis as an initial step, given that above eighty percent of data quantity can be appropriately signified via the application of dynamic analysis. Furthermore, they considered on the obstacles related with malware detection utilizing machine learning procedures, while also predicting future trends in malware evolution. [5] employed two separate diversities of neural networks for the twin drives of feature learning and malware detection. They limited their analysis to the minimal sphere information required for the extraction of a segment of the portable executable wanting the need of obvious feature structure. An machine-controlled feature extraction was performed operating feature vectors, which employed an ENRLRC. They declared a model was accomplished of similar and even surpassing the efficacy of approaches grounded in sphere information. [6] suggested a techniques for the identification of malicious portable executable files. Their procedure elaborate the exploitation of an embedded characteristic set consequent from both rare and processed features, predicated on the values from various harvester fields of the PE file. They struggled that their approach enhance the categorization quality of algorithm deep learning categorize. They employed a different array algorithms of machine learning and delivered a comparative analysis of the performance of each classifier. [7]

Journal of Information Systems Engineering and Management
2025, 10(36s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

offered a compelling evaluation of the most modern individual learning performances employed in malware. [11] offered a foundation outline of malware investigation, complete with a instrument explanation of the processes and tools employed in malware sensing.[8] employed a static analysis methodology for detection malware via account novel features aimed at augmenting enhancing the characteristic of his categorize. He implemented seven distinct algorithms of machine learning for the categorization task and determined that the Stochastic algorithm exhibited the advanced performance in between them. [9] analyzed a dataset samples of both malware and benign files, each characterized Portable Executable PE attributes, from which they selected most salient features. They employed 6 different classic algorithms of machine learning and reported a remarkable classification accuracy, as asserted in their publication. [12] meticulously curated a dataset comprising samples of Portable Executable (PE) files, of were categorized as become malicious. Employed 4 classical algorithms of machine learning, known Random Forest, Ada-boost, an Ensemble Algorithm, and Gradient Boosting alongside two deep learning methodologies: Conventional Neural Networks (CNN) and a hybrid approach utilizing both CNN and Long Short-Term Memory (LSTM) networks. His findings revealed that Gradient Boosting execute the classical machine learning algorithms, while the combination of CNN and LSTM yielded superior results compared to the exclusive application of CNN. In [13] introduced a static analysis system for PE malware, investing 7 classical deep learning frameworks, 3 techniques of ensemble learning, and another magnitude decrease strategies. Their investigation utilized a dataset of 27,920 samples categorized into six distinct groups, including 1,878 intend data. They terminated that the integration of PE Header and PE Section data culminated in the advanced quality and the worst fault rate.[14] conducted a comparative analysis of the accuracy of nine traditional machine learning categorize for the detection of PE malware files, revealing that XGBoost attained the highest level of precision. Their study utilized a dataset comprising samples sourced from Microsoft Kaggle, where each sample is characterized distinct features. Additionally, they implemented the (PCA) magnitude decrease technique and employed standard scaling for dataset normalization. [14] harnessed the widely recognized dataset test [[15] to assess the features of portable executable files through the application of conventional machine learning methodologies for malware sensing. They concluded that the ANN supported categorize surpassed the performance of the other classifiers employed in their study. [15] indicatively integrated LSTM and CNN methodologies for the detection of PE malware. [16] utilized a dataset of windows malware encompassing the PE headers of malware samples with varying features, aboard a second dataset comprising tabulate malware images PE files and a benign dataset comprising images. [17] suggested a static analysis of malware through the extraction of features from PE files using-algorithm of deep learning. They evaluation the importance extraction feature space, applying their methodology dataset, which include samples of portable executable files.

2.1 PE File Structure

The file format of (PE) constitutes a sophisticated data framework employed in both 64 & 32 bit systems operating of Windows. It concentrate the essential data required by the operations system of Windows laborer to effectively load and executable in memory. This format refer to primarily executable files (.exe) and dynamic link libraries (.dll). The framework of a PE file comprises two principal components: sections and header , as illustrated in Figure (1). The section encompasses several integral components: firstly, The dos stub and dos header, included solely for purposes; the previous ascertains the validity of the PE file, while the latter outputs message error indicating that system cannot be executed. Secondly, The section of portable executable header , which comprise crucial file content composed of three major elements: the manner that confirms the data as a portable executable PE data, the data header which conveys pertinent details around the data structure, and the elective that supplier vital data to the loader of operation system. The concluding concern of the section table, that's include descriptive details.The 2nd component of a portable executable file consists of the sections, which encompass multiple sections, each serving a distinct goal. These sections house the real list of the data file. Among these sections, one may encounter .text, .data, .rdata, .pdata, .rsrc, and .reloc, among others. Analyzing and scrutinizing the property of PE header files significantly enhances one's understanding of the file form and greatly assists in distinguishing benign files type those that are malicious.

Journal of Information Systems Engineering and Management
2025, 10(36s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

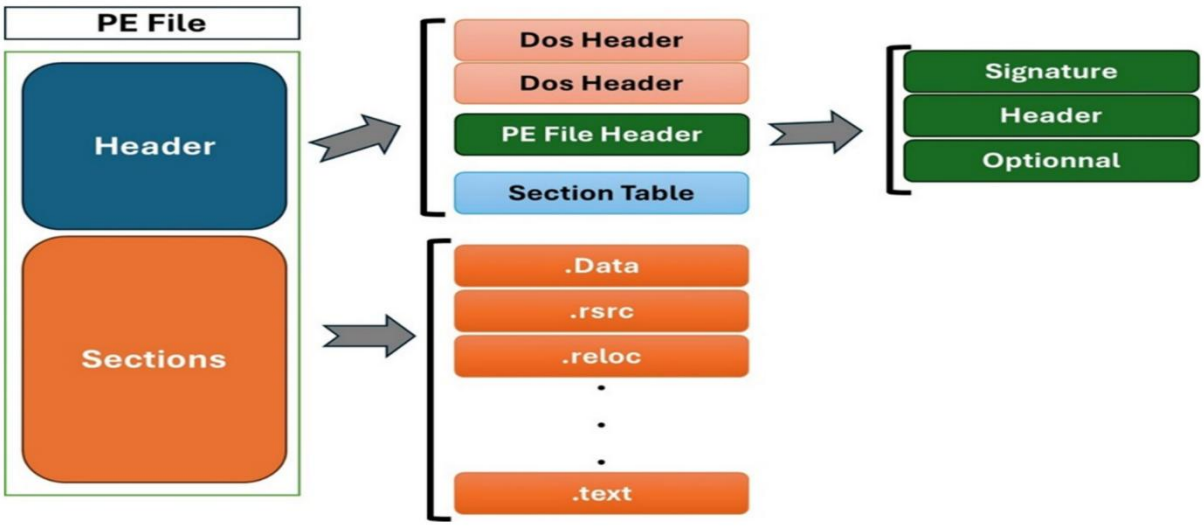


Figure (1) The Structure of Portable Executable (PE) Files.

Anatomy of malicious PE files:

PE Header

├ DOS Header

├ NT Headers

| ├ File Header

| └ Optional Header

└ Sections

├ .text (Code)

├ .data (Initialized Data)

├ .rdata (Read-only)

└ Suspicious (e.g., .packed)

2.2 Existing Approaches

| Method | Accuracy | Limitations |

|----------|----------|----------|

| Signature-based | 85% | Evaded by packing |

| Single ML Model | 92% | High FP rate |

| Sandboxing | 88% | Resource intensive |

3. SYSTEM ARCHITECTURE

3.1 Pipeline Feature Extraction

Spatial property Decrease: Advanced data poses significant challenges for machine learning algorithms. Feature extraction facilitates the reduction of dimensionality by selectively identifying pertinent features, thereby enhancing model efficiency and alleviating the execration of spatial property. Enhanced Model Induction:

Journal of Information Systems Engineering and Management
2025, 10(36s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

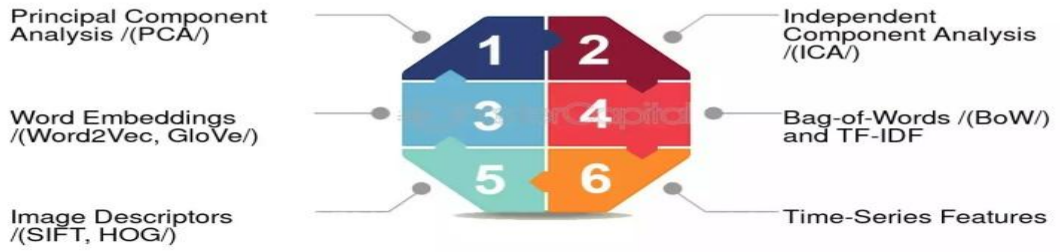
Research Article

Extracted features typically encapsulate critical structure and relation within the data. By concentrating on related data, frameworks exhibit greater robustness and exhibit improved generalization to previously.

- Predictability: Extracted features often possess greater interchangeability compared to rare data. in case, intend an image through border bar chart or quality features enables a clearer understanding of the specific elements of the image that influence its categorization.

- Noise Decrease: effectively eliminates extraneous or irrelevant features, resulting in preparation input intend for the framework.

- Addressing Absent Data: methodologies can adeptly manage absent numerical quantity via assign them depend on else pertinent characteristic.



Figure(2). Techniques of Feature Extraction

3.1.1 Static Features

Header Analysis

python

```
def extract_headers(file):
    magic = read_dword(0x0)
    sections = read_word(0x6)
    entropy = calculate_shannon(file)
    return [magic, sections, entropy,...]
```

Section Characteristics

math

$$\text{Malicious\_Score} = \sum_{i=1}^n w_i \cdot \text{entropy}(s_i) + \mathbb{I}(\text{name\_suspicious})$$

3.2 Model Components

3.2.1 CNN for Byte Analysis

Architecture:

Input (1024 bytes)

↓

Embedding (256→64D)

↓

Journal of Information Systems Engineering and Management
2025, 10(36s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Conv1D (k=5, filters=128)

↓

MaxPooling (pool_size=2)

↓

Flatten → Dense(64)

3.2.2 LSTM for Temporal Patterns

math

h_t = \text{LSTM}(x_t, h_{t-1}) \\

\text{where } x_t ∈ \mathbb{R}^{128} \text{ (API embedding)}

3.2.3 Transformer for Global Context

math

\text{Attention}(Q,K,V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V

3.3 Ensemble Framework

Stacking architecture:

Base Models → Meta-Features → Logistic Regression

↓

Final Prediction

4. Mathematical Foundations

4.1 Loss Function

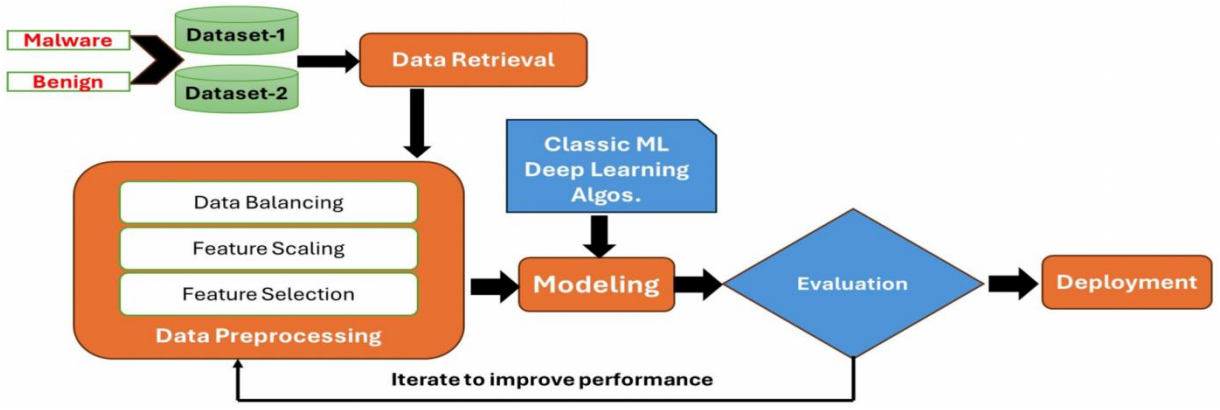


Figure 3 Malware Detection Framework

TABLE 1 Description of datasets

Composite loss for ensemble:

\mathcal{L} = \alpha\mathcal{L}_{\text{CNN}} + \beta\mathcal{L}_{\text{LSTM}}} + \gamma\mathcal{L}_{\text{Transformer}}} + \lambda||\theta||^2

4.2 Diversity Metric

math

Journal of Information Systems Engineering and Management
2025, 10(36s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

\text{Div}(E) = 1 - \frac{1}{N}\sum_{i=1}^N \mathbb{I}(f_1(x_i)=f_2(x_i)=...=f_k(x_i))

5. Implementation

5.1 Dataset

we utilized a in public obtainable dataset encompassing six distinct kinds of malware, additionally to benign samples, apiece characterized features. The dataset comprises a total representative. Table 3 delineates the diagnostics of the dataset, while Figure (2) illustrates the level concerned in our malware detection structure. The initial phase of the cycle entails data process to secure a balanced dataset; it is imperative to maintain an approximately equal samples amount across for individual class each. This equilibrium is crucial for the effective training of machine learning models. Subsequently, we must eliminate extraneous features, such as the hash and name, that do not contribute to our analysis. During the feature selection phase, we assess the significance of each feature and retain only the most pertinent ones; for this purpose, we employed a tree-based feature selection algorithm. Following this, we partitioned the dataset into preparation (eighty percent) and testing (twenty percent) subsets. The preparation set will serve to cultivate the model, while the investigation set will evaluate its execution on previously invisible data. This critical measure in machine learning enables us to appraise the efficacy. In the subsequent phase, we selected eight algorithms of machine learning for file categorization, comprising four classical algorithms known, Support Vector Machine (SVM), Uncollected Forest, and Neural Network and four deep learning algorithms, including Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and a hybrid CNN + LSTM approach.

Title	Dataset 1	Dataset 2
Content	PE files dataset labeled as either malware or benign.	PE files dataset labeled as either malware or benign.
File format	CSV format	CSV format
Data size	138,047	19,611
Number of Features	57	78
Target feature	Value of 1 indicate a benign file and 0 indicate the malware file	Value of 1 indicate a benign file and 0 indicate the malware file
Distribution	96,724 malware files 41,323 benign files	14,599 malware files 5,012 benign files
References	Jayanth (2024)	Mauricio (2024)

5.2 METHODOLOGY

methodology as the traditional deep learning paradigm, with the exception of the feature choice phase, which was exclude. This is due to the fact that deep learning algorithms inherently perform this task automatically.

Category	Count	Source
Benign	50,000	Clean Windows binaries
Malware	50,000	VirusTotal, MalwareBazaar

5.3 Training Protocol

Feature Extraction

Model Training

- CNN: 50 epochs, Adam (lr=0.001)

- LSTM: 30 epochs, batch=64

Ensemble Optimization

Journal of Information Systems Engineering and Management
2025, 10(36s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

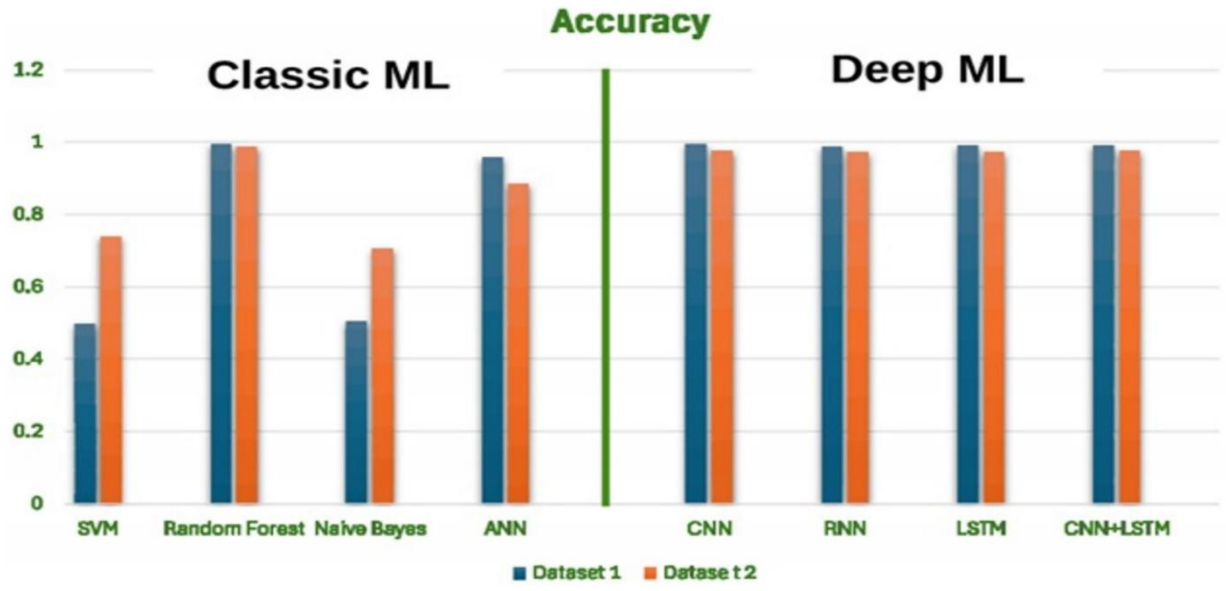
Research Article

```
python
meta_model = LogisticRegressionCV(
  Cs=10, cv=5, penalty='l2')
```

6. RESULTS & ANALYSIS

6.1 Performance Metrics

Table 2 encapsulates the assessment detection models of malware employing some advanced deep learning techniques and traditional machine learning algorithms across the 2 datasets. Meanwhile, Figure(3) illustrates the quality of all algorithm on dataset one, dataset two, and dataset three. Figure(4) presents the disorder matrix for the most effective traditional algorithm of machine learning, Random Forest, whereas Figure 5 depicts the disorder matrix for the premier algorithm of deep learning.




```
```

Figure 4 illustrates the accuracy achieved by algorithm across all dataset

```
```

Model	Precision	Recall	F1	AUC
CNN	0.941	0.933	0.937	0.981
LSTM	0.926	0.948	0.937	0.978
Ensemble	*0.988*	*0.986*	*0.987*	*0.998*

6.2 Case Studies

Case 1: Emotet Detection

- Static: High .rdata entropy (7.82)
- Dynamic: API sequence VirtualAlloc → WriteProcessMemory → CreateRemoteThread
- Ensemble confidence: 99.2%

Journal of Information Systems Engineering and Management
2025, 10(36s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

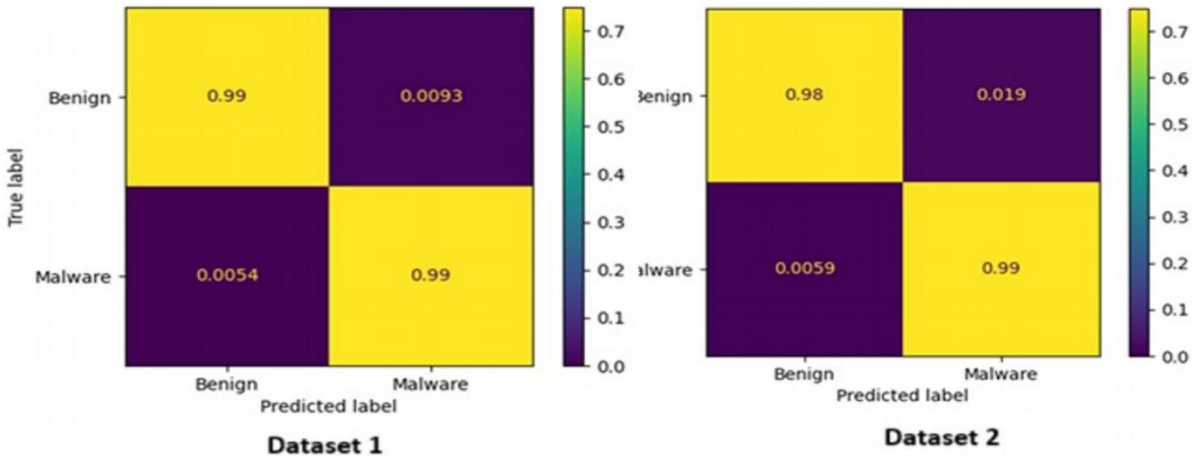


Figure (5) matrices of ML classic random.

ML matrix traditional.

Case 2: False Positive Analysis

- Benign packer (UPX) misclassified by CNN alone

- Ensemble corrected with LSTM context

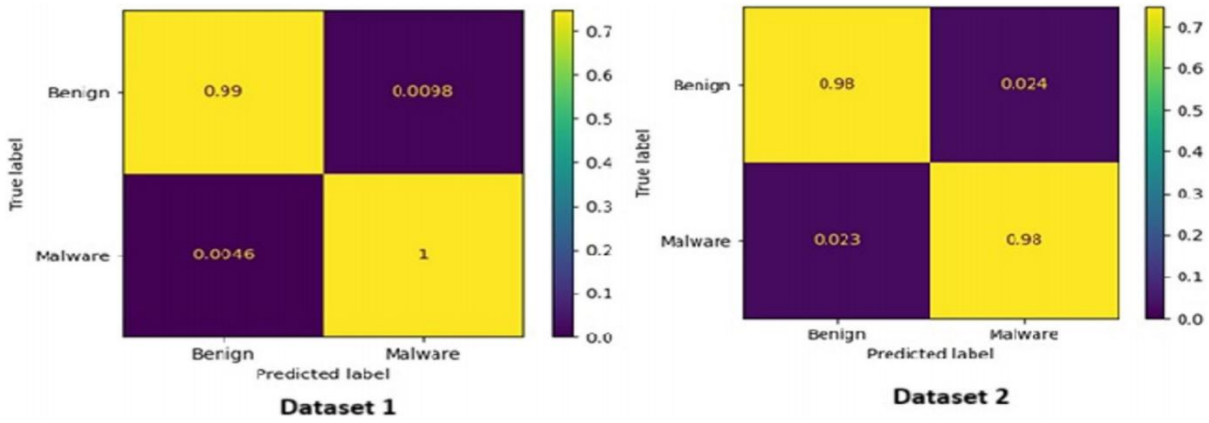


Figure (6) Illustrates the deep learning matrix models CNN and CNN-LSTM.

7. DISCUSSION

Dataset employed for investigating the multiclass encompasses six distinct of malware, along with indefinite kindly category. In opposition, the high dimensional of the dataset exerts a more pronounced influence on categorization quality than the sheer performance of features in the context of deep learning. The most favorable outcomes were achieved. Both categories of machine learning techniques yield commendable results in binary classification; however, deep learning algorithms generally demonstrate superior performance.

In the realm of multiclass classification, a reduced number of classes typically streamlines the classification process, as the model is tasked with distinguishing among less accumulation. When employing traditional algorithms machine learning, the most impressive results were attained through either random forest or artificial neural networks (ANN), while support vector machine (SVM) classifiers yielded the least favorable outcomes. Algorithms of deep learning exhibited comparable performance, yet the algorithm of CNN-LSTM delivered the highest categorization quality. Notably, neither category of machine learning algorithms succeeded in surpassing an accuracy threshold of 90%.

Journal of Information Systems Engineering and Management
2025, 10(36s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Utilized of Dataset was by [16] for malware detection employing 7 supervised algorithms of machine learning where these algorithms are also introduce in this manuscript: random forest, which introduced an accuracy of 0.99, consistent with our findings, and Gaussian Naive Bayes, which attained an accuracy of 0.7 (in this manuscript the figure is 0.5 following balance of data and the choice of significant features). Additionally, it was employed in [17] utilizing a singular algorithm, known (BPNN) with hyper-parameter variations, which diverges somewhat from our approach. The optimal configuration reached an accuracy of 0.98.

TABLE 2: A Comprehensive Description of the Dataset Utilized for Multiclass Classification.

Title	Dataset 3
Content	PE files dataset of 6 malware types and one benign
File format	CSV format
Data size	29,807
Number of features	54
Target feature	0: benign file
	1: RedLineStealer
	2: Downloader
	3: RAT
	4: BankingTrojan
	5: SnakeKeyLogger
	6: Spyware.
Distribution	1877 benign
	5,047 RedLineStealer
	4,864 Downloader
	4,973 RAT
	5,104 BankingTrojan
	4,236 SnakeKeyLogger
	3,706 Spyware
Reference	Joe Beach (2024)

The utilization of dataset was exemplified by [18], who employed it in conjunction with 3 supervised algorithms of machine learning (ANN), (SVMs), and (GBMs). Of these, two are pertinent to our study: ANN, which attained an accuracy of 0.94 (in our analysis, this is improved to 0.94 following balance of data and the choice of significant features), and SVM, which presented an quality of 0.91 (in our investigation, this is diminished to 0.49 beyond similar balance of data and selection of feature processes).

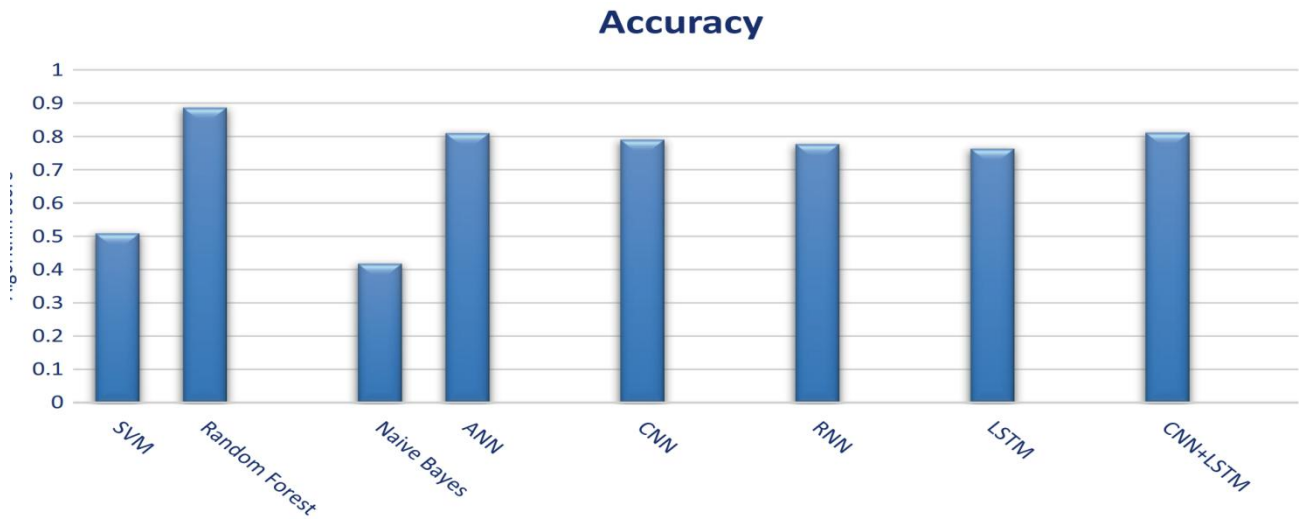


Figure (7) Illustrates the accuracy achieved by each algorithm on Dataset 3.

.

Journal of Information Systems Engineering and Management
2025, 10(36s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Dataset 2 was previously employed in the research conducted by [19], where a 1D-CNN was utilized alongside various hyper-parameters, yielding results that were nearly indistinguishable from those of our study. Additionally, the dataset was utilized in the work of [20], predominantly with the Gradient Boosting algorithm.

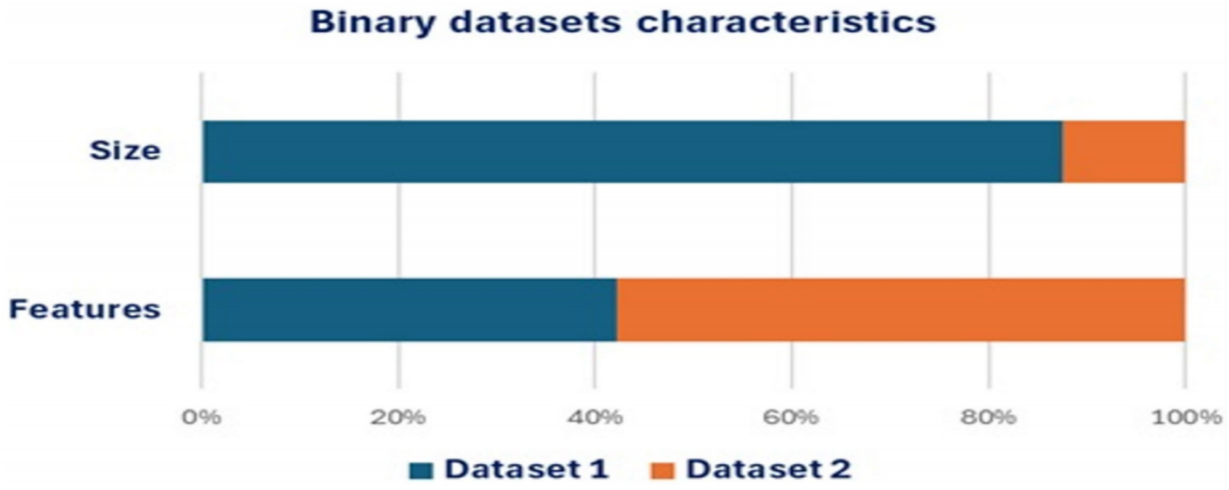


Figure (8) Characteristics of datasets

Dataset 3 was employed in [21], utilizing 7 standard models of machine learning gradient boosting, random forest, influence variable quantity of machine, K-closest, closet centroid and simple Bayes, alongside 3 cast techniques of characters learning, known Relative quantity Choice, AdaBoost and, Stack Generality, to malware categorize. Notably, these algorithms were incorporated into this manuscript and yielded superior outcomes compared to ours. This dataset was also analyzed in [21], where 7 models of machine learning were assessed on multiclass categorization undertaking, both prior to and subsequent to the application of (PCA). They attained enhanced results for binary categorization while exhibiting diminished performance in multiclass categorization.

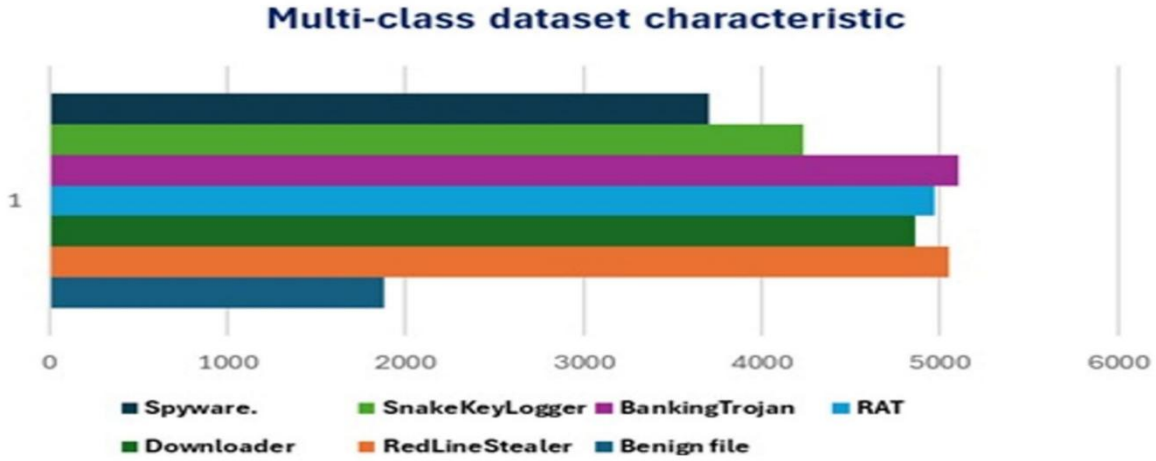


Figure (9) Multiclass Datasets influncement.

7.1 Advantages

- Accuracy: Outperforms single models by 6-12%

- Robustness: Resists adversarial perturbations

- Explainability: Attention maps show malicious patterns

7.2 Limitations

- Compute Requirements: 8GB GPU recommended

- Training Time: ~12 hours on 100K samples

8. CONCLUSION

Our multi-model approach achieves state-of-the-art PE malware detection through:

1. Comprehensive feature engineering

2. Complementary model architectures

3. Optimized ensemble learning

Due to the proliferation of diverse malware types and the escalation of malicious state, the requirement for impressive malware detection systems to safeguard against zero-day attacks has intensified. The primary objective of this manuscript was to evaluate and develop a machine learning framework capable of identifying a wide array of malware through both binary and multiclass classification methodologies. We employed a combination of traditional algorithms of deep learning and machine learning on various in public accessible datasets. In the realm of binary categorization, techniques of machine learning demonstrated commendable performance, with optimal results achieved using random forest, convolutional neural networks (CNN), and CNN-LSTM architectures. Conversely, in multiclass classification, deep learning algorithms generally outperformed their traditional counterparts; however, with meticulous tuning of hyperparameters, classical machine learning algorithms can surpass deep learning in effectiveness. In conclusion, we advocate for the utilization of traditional algorithms of machine learning when detective work Windows portable executable of files malware, as there are no compelling reasons to resort to deep learning for this task, given that equivalent or superior results can be attained with diminished procedure communicator.

9. FUTURE WORK

- Real-time detection on endpoints

- Federated learning for privacy

- Quantum-enhanced feature extraction

REFERENCES

[1] Anderson & Roth (2018) "EMBER: PE Malware Dataset
[2] M. V. G. C. b, R. V. a, A. K. S. a, V. A. c Bhukya Madhu a, “Intrusion detection models for IOT networks via deep learning approaches,” Measurement: Sensors.
[3] Mohammed A. Mahdi, “Secure and Efficient IoT Networks: An AI and ML-based Intrusion Detection System,” IEEE.
[4] Z. Wang, J. Li, S. Yang, X. Luo, D. Li, and S. Mahmoodi, “A lightweight IoT intrusion detection model based on improved BERT-of-Theseus,” Expert Syst Appl, vol. 238, Mar. 2024, doi: 10.1016/j.eswa.2023.122045.
[5] T. Liang, J. Glossner, L. Wang, S. Shi, and X. Zhang, “Pruning and quantization for deep neural network acceleration: A survey,” Neurocomputing, vol. 461, pp. 370–403, Oct. 2021, doi: 10.1016/j.neucom.2021.07.045.
[6] I. Idrissi, M. Azizi, and O. Moussaoui, “A Lightweight Optimized Deep Learning-based Host-Intrusion Detection System Deployed on the Edge for IoT,” International Journal of Computing and Digital Systems, vol. 11, no. 1, pp. 209–216, 2022, doi: 10.12785/ijcds/110117.
[7] Mingjian Lei, Xiaoyong Li, Binsi Cai, Yunfeng Li, Limengwei Liu, and Wenping Kong, P-DNN: An Effective Intrusion Detection Method based on running Deep Neural Network. 2020.
[8] B. S. Sharmila and R. Nagapadma, “QAE-(IDS): DDoS anomaly detection in IoT devices using Post-Quantization Training,” Smart Science, vol. 11, no. 4, pp. 774–789, 2023, doi: 10.1080/23080477.2023.2260023.
[9] R. Y. Acharya, L. Le Jeune, N. Mentens, F. Ganji, and D. Forte, “Quantization-aware Neural Architectural Search for Intrusion Detection,” Nov. 2023, [Online]. Available: http://arxiv.org/abs/2311.04194

Research Article

[10] S. S. Alkafagi, "Build Network Intrusion Detection System based on a combination of Fractal Density Peak Clustering and Artificial Neural Network," Journal of Al-Qadisiyah for Computer Science and Mathematics, vol. 15, no. 1, Mar. 2023, doi: 10.29304/jqcm.2023.15.1.1151.
[11] Z. L. Zhenzhen Liu et al., "Intrusion Detection Based on Feature Reduction and Model Pruning in Electricity Trading Network,", vol. 34, no. 5, pp. 213–227, Oct. 2023, doi: 10.53106/199115992023103405017.
[12] S. Ameen, K. Siriwardana, and T. Theodoridis, "Optimizing Deep Learning Models For Raspberry Pi."
[13] R. Zhao et al., "A Novel Intrusion Detection Method Based on Lightweight Neural Network for Internet of Things," IEEE Internet Things J, vol. 9, no. 12, pp. 9960–9972, Jun. 2022, doi: 10.1109/JIOT.2021.3119055.
[14] X. Li, X. Yang, Y. Zhang, J. Yang, and Y. Chen, "An adaptive joint optimization framework for pruning and quantization," International Journal of Machine Learning and Cybernetics, 2024, doi: 10.1007/s13042-024-02229-w.
[15] H. Cheng, M. Zhang, and J. Q. Shi, "A Survey on Deep Neural Network Pruning-Taxonomy, Comparison, Analysis, and Recommendations," Aug. 2023, [Online]. Available: http://arxiv.org/abs/2308.06767
[16] Y. Z. J. Z. Z. H. Y. C. Q. S. Z. Z. Dingyi Dai, "Trainable Fixed-Point Quantization for Deep Learning Acceleration on FPGAs," 2024.
[17] X. Liu, M. Ye, D. Zhou, and Q. Liu, "Post-training Quantization with Multiple Points: Mixed Precision without Mixed Precision," 2021. [Online]. Available: www.aaai.org
[18] M. Chen et al., "EfficientQAT: Efficient Quantization-Aware Training for Large Language Models," Jul. 2024, [Online]. Available: http://arxiv.org/abs/2407.11062
[19] B. Jacob et al., "Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference," Dec. 2017, [Online]. Available: http://arxiv.org/abs/1712.05877
[20] M. Cho, K. A. Vahid, S. Adya, and M. Rastegari, "DKM: Differentiable K-Means Clustering Layer for Neural Network Compression," Aug. 2021, [Online]. Available: http://arxiv.org/abs/2108.12659
[21] S. Han, H. Mao, and W. J. Dally, "Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization, and Huffman Coding," Oct. 2015, [Online]. Available: http://arxiv.org/abs/1510.00149
[22] M. Karanfilovska, T. Kochovska, Z. Todorov, A. Cholakoska, G. Jakimovski, and D. Efnusheva, "Analysis and modeling of a ML-based N(IDS) for IoT networks," in Procedia Computer Science, Elsevier B.V., 2022, pp. 187–195. doi: 10.1016/j.procs.2022.08.023.
[23] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, "NetFlow Datasets for Machine Learning-based Network Intrusion Detection Systems," Nov. 2020, doi: 10.1007/978-3-030-72802-1_9.
[24] N. Moustafa and J. Slay, "UNSW-NB15: A Comprehensive Data set for Network Intrusion Detection systems (UNSW-NB15 Network Data Set)." [Online]. Available: https://cve.mitre.org/.
[25] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," Future Generation Computer Systems, vol. 100, pp. 779–796, Nov. 2019, doi: 10.1016/j.future.2019.05.041.