An Explainable and Robust Machine Learning Framework for Polymorphic Malware Detection

*U Abhiram Patel[1], Lakshmi Narayan P[1], Supriya Goel[1], Pradeepthi K V[1]

[1] Department of Computer Science, C.R.Rao Advanced Institute of Mathematics, Statistics and Computer Science (AIMSCS), Hyderabad, India
Mail ID: *abhirampatel@crraoaimscs.res.in, lakshminarayan@crraoaimscs.res.in, supriya@crraoaimscs.res.in, pradeepthi@crraoaimscs.res.in

Abstract

Polymorphic Malware analysis has become a critical problem in Cyber Security and application of Machine Learning to the same is proving to be very useful as traditional signature based methods are failing. The malware is evolving continuously, so the machine learning algorithms being used should be good at generalizing. In this paper, we have used three dataset of real-world malware samples, Microsoft Malware dataset, DikeDataset and Malware Opcodes-Virus Share dataset. Random Forest algorithm is able to give 82 % accuracy and XGboost is giving 85 % accuracy, Performance evaluation using various cross-validation techniques was performed. The SHAP explainability algorithm was also applied to understand which features are contributing better for model performance. Our proposed algorithm is able to generalize well and when tested on unseen data, provides good accuracy.

Keywords: Malware Analysis, Polymorphic Malware, Machine Learning, Feature Extraction, Explainability, SHAP.

1. Introduction:

Malware has always been a huge concern for any industry that is internet facing, from technology firms, healthcare, manufacturing, education, or the various sectors in government like, electric power plants, nuclear power plants, smart grids, and many more. With growth in technology, malware industry too has evolved over time and it has become a bigger threat than anticipated. The initial malware programs though powerful, had limited range of signatures and behaviours. Now, the scenario has changed completely, malware have become polymorphic and have the ability to transform themselves into a new malware program with a brand new signature and characteristics. The new malware has minimal similarities with respect to the parent malware. Hence, the rule based and signature-based systems are unable to detect them.

We are taking the help of methodologies that are prevalent in the current era, for identification of intricate patterns in huge amounts of data. From literature, it is evident that technologies like Artificial Intelligence, Machine Learning, Deep Learning etc are aiding for developing more

D. R. Reddy et al. (eds.), *Proceedings of the First International Conference on Advances in Forensics and Cyber Technologies (ICFACT 2025)*, Advances in Computer Science Research 127,
https://doi.org/10.2991/978-94-6239-610-4_41

robust security systems. With the evolution of machine learning, identifying and classifying a malware has become more feasible. Cases of Polymorphic virus identification have also seen considerable amount of success.

However, despite the progress, there is still a high amount of False Positive Rate when it comes to identifying unknown malware signatures. Researchers have designed and conducted experiments to identify various malware signature using different pattern recognition approaches.

In this paper, we take data from three datasets, namely, Microsoft Malware dataset, DikeDataset and Malware Opcodes Virus Share dataset. Then we apply the various Machine Learning and Deep Learning algorithms and scrutinize the results. We have also applied PCA(Principal Component Analysis) to understand the spread of the data and visualize the malware families. The same is elaborated with adequate results analysis in the Section 5.

2. Literature Survey:

With technology, malware tools have evolved over time and are becoming more powerful and evasive. Though initial malware programs were powerful, however they lacked variation of signatures and behaviours. Now, the scenario has totally changed, malware programs have become polymorphic and can completely transform themselves into a new malware program with a brand new signature and characteristics. The evolution of machine learning and its applications to various pattern recognition problems is largely being used in identifying and classifying a malware. However polymorphic malware are a totally different ball game. Here new malware are being generated from existing malware, and knowing the signature of the Parent malware is not going to aid in recognising the Child Malware in any way. This makes this problem more intriguing. We have studied various papers from this domain. We have compared the various papers and mentioned the Malware analysed, Analysis Type, Technique Used, Key Contribution and Limitations in Table 1.

Table 1. Literature Survey

Ref.	Year	Malware Focus	Analysis Type	Technique Used	Key Contribution	Limitation
[1]	2025	Polymorphic Ransomware	Dynamic + DL	Deep Neural Networks	High detection accuracy for unseen ransomware variants	High computational overhead
[2]	2024	Polymorphic Malware	Hybrid	ML + Behavioral Analysis	Comprehensive framework combining static & dynamic traits	Complex feature extraction
[3]	2024	Polymorphic Virus	Static	Advanced ML Models	Improved identification of	Limited adaptability to

					polymorphic variants	zero-day malware
[4]	2024	General Malware	Survey	ML Techniques	Trends and challenges in ML-based malware detection	No experimental validation
[5]	2024	Polymorphic Malware	Dynamic	Lightweight Ensemble (Federated)	Edge-friendly detection with low overhead	Limited dataset diversity
[6]	2025	Polymorphic Botnets	Survey	ML-based Defenses	Overview of evolving botnet detection strategies	Lack of implementation results
[7]	2024	Malware (General)	Static	Supervised ML	Demonstrates effectiveness of ML classifiers	Struggles with obfuscation
[8]	2024	Malware (General)	Static	ML Algorithms	Enhanced classification accuracy	Limited focus on polymorphism
[9]	2022	Malware (General)	Static	ML Algorithms	Comparative ML evaluation	Poor performance on evasive malware
[10]	2025	Polymorphic Malware	Hybrid	Hybrid ML Algorithms	Combines multiple feature domains	Increased model complexity
[11]	2021	Polymorphic & Metamorphic Malware	Conceptual	Theoretical Analysis	Foundational understanding of malware evolution	No detection framework
[12]	2024	Polymorphic Ransomware	Dynamic	GAN + DL	Synthetic sample generation for novel malware	Training instability
[13]	2022	Malware Variants	Dynamic	Deep Ensemble Models	Robust behavioral detection	High resource consumption
[14]	2024	Zero-day Malware	Behavioral	ML-based Detection	Effective detection of first-seen malware	Requires large training data
[15]	2021	Malware	Static	CNN +	Improved	Limited

				Gradient Boosting	feature learning	polymorphic focus
[16]	2021	Malware Types	Survey	ML Classification	Comprehensive taxonomy of ML approaches	Outdated datasets
[17]	2022	Windows Malware	Static	Deep Neural Networks	Strong performance over classical ML	OS-specific
[18]	2023	Polymorphic Malware	Hybrid	Analysis Model	Conceptual polymorphic detection model	No real-world evaluation
[19]	2021	Polymorphic Malware	Hybrid	Multi-Faceted Detection	Improved detection accuracy	High system complexity
[20]	2021	Windows PE Malware	Static	Ensemble ML + NN	Reduced false positives	Limited generalization

From the literature we can understand that most of the works have not taken real-world malware samples for analysis and the developed frameworks have not generalized well. In this paper we attempt to solve both these problems and explore explainability as well.

3. Methodology:

Polymorphic Malware alter their internal structure to evade signature-based detection, while preserving their malicious behaviour. This work proposes a static analysis based malware classification framework using opcode frequency analysis, based on the features extracted from disassembled assembly files. Opcode unigram distribution, entropy measures and interaction-based bigram features are used to represent malware behaviour. Multiple Machine Learning classifiers are evaluated on the Microsoft Malware dataset, DikeDataset and Malware Opcodes Virus Share dataset. Experimental Results are show in the subsequent sections, where Random Forest classifiers achieve robust performance across malware families, with strong generalization validated through stratified cross-validation and hyper parameter optimization.

Now let us look at the various components of the proposed methodology in detail. The schematic diagram in Figure 1 details out the various steps.

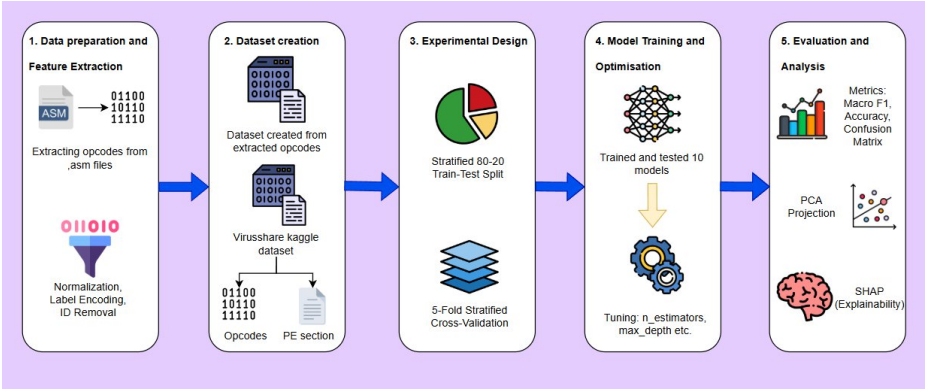

Fig. 1. System Architecture.

3.1 Dataset Preparation:

The three datasets used for our analysis are described here. In the Microsoft Malware Classification Challenge (BIG 2015)[21] , there are over 20,000 malware samples grouped into nine families, presented as hexadecimal dumps and disassembled assembly code minus the executable headers. It is one of the most preferred datasets for training and testing machine learning and deep learning models for malware family classification and static analysis.

DikeDataset[22] is a labeled collection of harmless and malicious executables, specifically designed for malware detection research. Coming from the real world, it is comprised of genuine binary files that can be used for static, dynamic analysis, and basically any binary-based classification experiments.

The Malware Opcodes Virus Share [23] dataset is made up of opcode sequences from malware programs, which are used to identify and classify malware. This abstraction of raw binaries into a series of opcodes makes it possible to use machine learning-based analysis.

3.2 Data Preparation and Preprocessing and Data Creation:

The dataset have the executable files, from which the opcodes are to be extracted and analysed. These opcodes will then need to be further processed to generate the dataset that is Machine Learning ready. Firstly the opcode are further processed and Unigram analysis is performed on it. From the analysis, the unigram frequency features of the Opcodes are extracted. Along with the Opcode features, to add more strength to the features, we also ass PE section indicators, Register usage statistics and Opcode entropy to the dataset. Combining all these

features to form the dataset was one critical step, which eventually led to the development of a good pattern recognition system. Because, if the data being fed into the machine learning algorithm is itself not having enough essence of the malwares that we are trying to understand, then even powerful algorithms will fail to find any patterns.

3.3 Experimental Design:
The dataset is split into 80% for training and 20% for testing, to allow the Learning algorithm to learn all the different examples. Microsoft Big dataset, has class imbalance problem, hence we employed class-weighted classifiers. For this dataset, Random Forest algorithm was used as the base model due to its robustness to feature correlation, noise and non-linear decision boundaries.

3.4 Model Training and Optimization:
To evaluate the performance of the different malware datasets, around then machine learning algorithms and deep learning algorithms were used. Ten fold cross validation was applied to the datasets to reduce over fitting. Optimization of the various learning parameters of the algorithm was also done at this stage to allow better learning rate, even for unseen data.

3.5 Evaluation and Analysis:
After all the algorithms were applied, we then measured the performance of the various algorithms by studying the various performance metrics like, Accuracy, Precision, Recall, Macro-averaged F1-score and Confusion Matrix. Of these, Macro- F1-score gives more emphasis to handling the class imbalance problem. The specific details of the algorithm results are discussed in the next section.

4. Results and Discussion:

The distribution of the polymorphic malware families are shown using PCA on the data. This is shown in Figure 2.

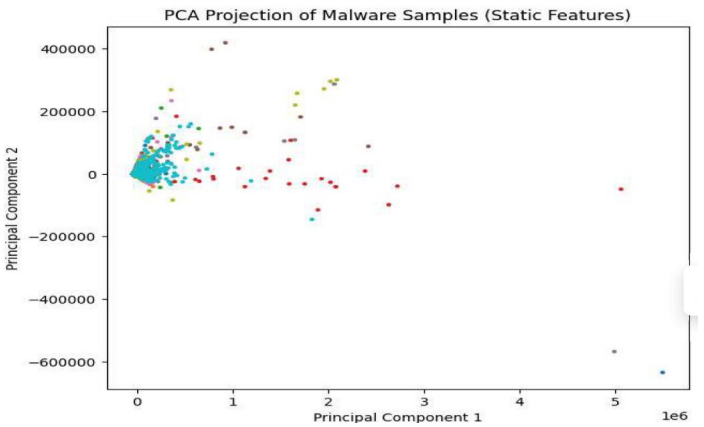

Fig. 2. PCA Visualization of the three datasets.

Multiple other algorithms were tested with the data of BIG dataset for understanding which machine learning algorithm is better suited. The results are shown in Table 2.

Table 2. Values of various evaluation metrics for different algorithms for BIG dataset

Model	Accuracy	Macro F1-Score
Naïve Bayes	0.61	0.48
Decision Tree	0.98	0.96
K-NN	0.96	0.88
Random Forest	0.99	0.99

Initial experimentation was performed on the Microsoft Big Dataset, where Random Forest was applied on the data.

Table 3. Values of various evaluation metrics of Random Forest Algorithm

Class	Precision Score	Recall	F1	Support
1	0.9871	0.9935	0.9903	308
2	1.0000	0.9980	0.9990	496
3	0.9966	0.9983	0.9975	588
4	0.9895	0.9895	0.9895	95
5	1.0000	1.0000	1.0000	8
6	0.9933	0.9933	0.9933	150
7	1.0000	1.0000	1.0000	80
8	0.9918	0.9837	0.9878	246
9	1.0000	1.0000	1.0000	203
Accuracy	0.9954			

From the table 3, we can conclude that Random Forest is giving good results for almost all the classes and the accuracy of 99% is achieved

To access the generalization and prevent overfitting, 5-fold cross validation is performed using Micro F1 score, whose value was obtained as 0.9866.

To understand how Random Forest results better, the Explainable AI algorithm of SHAP(Shapley Additive exPlanations). The below screenshots in Figure 3 show the SHAP output.

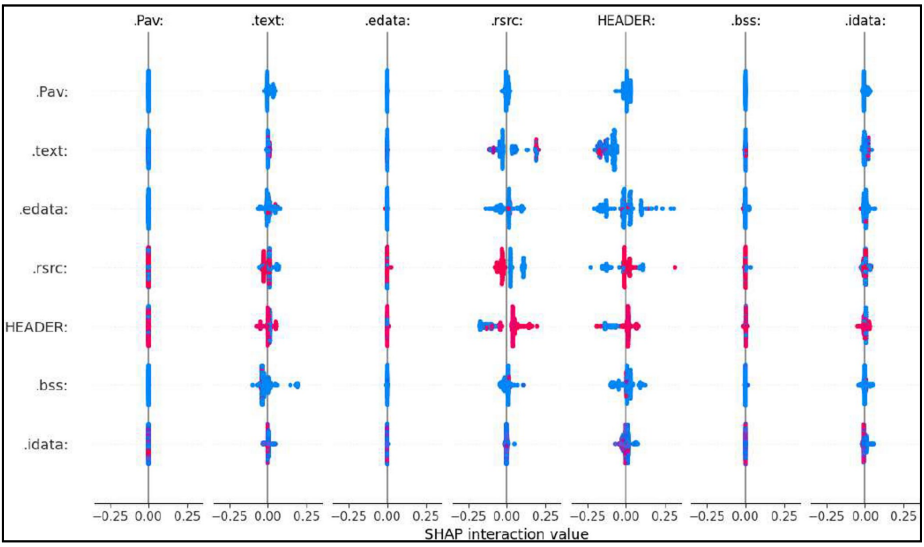

Fig. 3. Output of SHAP algorithm for BIG dataset

From the results of the SHAP algorithm, we can understand that opcode features influence the output more, like, the opcodes JAE, JNE, XOR, RET, have more influence that others. In PE .rs, .c, header, have better influence on the results.

Now, the three datasets considered in this work for polymorphic malware analysis, using Static opcode distribution and PE structural feature extraction, are subjected to Machine Learning algorithms, as shown in Table 4.

Table 4. Values of various models' evaluation metrics of our proposed framework

Model	Accuracy	Macro F1-Score
XGBoost	0.853470	0.76496
Random Forest	0.829820	0751612
KNN	0.782005	0.69489
Decision Tree	0.762982	0.678846
Extra Trees	0.753213	0.665303
SVM(RBF)	0.61538	0.573805
Logistic Regression	0.523933	0.474255
Naive Bayes	0.427249	0.336152
Wide-Deep MLP	0.605141	0.53323
MLP	0.601542	0.55325
Autoencoder+classifier	0.557326	0.52352

From the results in Table 3, we can understand that the algorithms in understand that compared to Deep Learning algorithms, the Machine Learning algorithms, especially that Random Forest algorithm performs better.

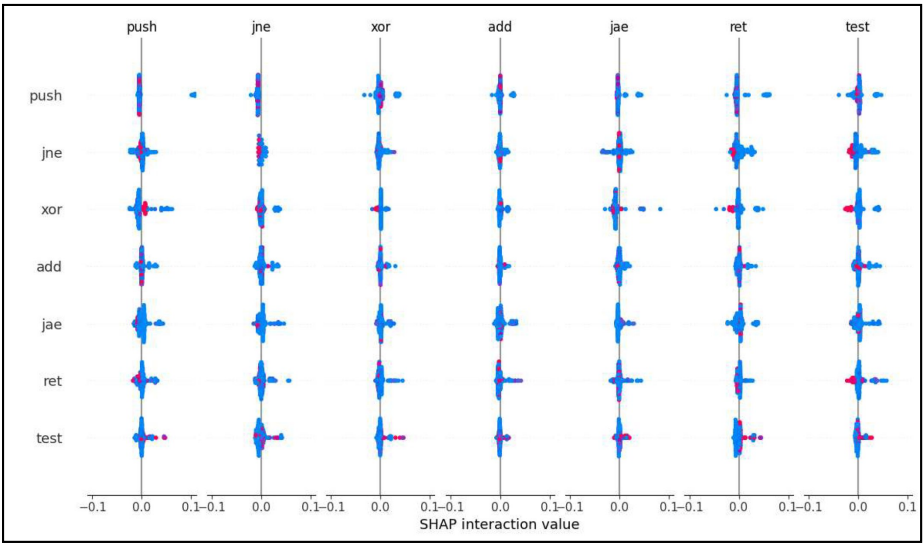

Fig. 4. Output of SHAP algorithm for Malware Opcodes Virus Share

To interpret the decisions of the Random Forest classifier, SHAP(SHapley Additive exPlanations) is employed. SHAP is able to quantify the contribution of the different Static Features based on the model predictions. The opcodes like PUSH, XOR and RET have higher contribution in Malware Opcodes-Virus Share dataset also.

5. Conclusion

The problem of Polymorphic malware analysis is very challenging and ever evolving due to the usage of technology and AI the adversary. The techniques have evolved so much that the parent and child malware have very less commonality and it has become almost impossible to find any similarity using signatures. At this juncture, this paper studies three real-world malware samples and performs a complete machine learning based analysis. Our results show that Random Forest algorithm is able perform very well for real world sample. We are also eliminating the problem of generalization and avoiding overfitting by using cross validation and hold-one out methods.

References:

[1] Gazzan, Mazen, Bader Alobaywi, Mohammed Almutairi, and Frederick T. Sheldon. "A Deep Learning Framework for Enhanced Detection of Polymorphic Ransomware." Future Internet 17, no. 7 (2025): 311.

[2] Chaikovskyi, Maksym, Inna Chaikovska, Tomas Sochor, Inna Martyniuk, and Oleksii Lyhun. "Comprehensive approach to the detection and analysis of polymorphic malware." In CEUR Workshop Proceedings, vol. 3736, pp. 312-323. 2024.

[3] Anil, D., R. Shreeshayana, and B. Kiran. "Advanced Malware Detection Methods for Polymorphic Virus Identification." In 2024 5th International Conference on Communication, Computing & Industry 6.0 (C2I6), pp. 1-6. IEEE, 2024.

[4] Polamarasetti, Anand. "Research developments, trends and challenges on the rise of machine learning for detection and classification of malware." In 2024 International Conference on Intelligent Computing and Emerging Communication Technologies (ICEC), pp. 1-5. IEEE, 2024.

[5] Mittal, Sakshi, and Prateek Rajvanshi. "Lightweight Ensemble Approach To Polymorphic Malware Detection with Dynamic Analysis on Federated Computational Edges."

[6] Ahmadi, Sina. "Evolving botnet defenses: A survey of machine learning approaches for identifying polymorphic and evasive malware." International Journal of Innovative Research and Scientific Studies 8, no. 2 (2025): 338-347.

[7] Abhijna, C. D., A. S. Aishwarya, B. R. Sai Pranam, and S. M. Raghuramegowda. "Malware detection using machine learning." In 2024 Second International Conference on Advances in Information Technology (ICAIT), vol. 1, pp. 1-5. IEEE, 2024.

[8] Kurian, Alen Jacob, Akshay Santhosh, and Midhun Subin. "Enhanced malware detection framework leveraging machine learning algorithms." International Research Journal of Modernization in Engineering Technology and Science 6, no. 03 (2024): 3597-3603.

[9] Akhtar, Muhammad Shoaib, and Tao Feng. "Malware analysis and detection using machine learning algorithms." Symmetry 14, no. 11 (2022): 2304.

[10] Hamadjida, Halilou Claude Bobo, Aurelle Tchagna Kouanou, Christian Tchapga Tchito, And Clarence Tamko Kouadjo. "Malware's Polymorphism Analysis Using A Hybrid Machine Learning Algorithm Approach." In Data Science Workshop From Theory To Practice, 2025.

[11] Badhwar, Raj. "Polymorphic and metamorphic malware." In The CISO's Next Frontier: AI, Post-Quantum Cryptography and Advanced Security Paradigms, pp. 279-285. Cham: Springer International Publishing, 2021.

[12] Arlowe, Matthew, David Fitzpatrick, and Christopher Brannon. "Evolutionary generative adversarial networks for detecting novel polymorphic ransomware." (2024).

[13] Al-Hashmi, Asma A., Fuad A. Ghaleb, A. Al-Marghilani, Abdulsamad E. Yahya, Shouki A. Ebad, Muhammad Saqib, and Abdulbasit A. Darem. "Deep-ensemble and multifaceted behavioral malware variant detection model." IEEE Access 10 (2022): 42762-42777.

[14] Shaukat, Kamran, Suhuai Luo, and Vijay Varadharajan. "A novel machine learning approach for detecting first-time-appeared malware." Engineering Applications of Artificial Intelligence 131 (2024): 107801.

[15] Thosar, Keshav, Pranay Tiwari, Revanth Jyothula, and Dayanand Ambawade. "Effective malware detection using gradient boosting and convolutional neural network." In 2021 IEEE Bombay Section Signature Conference (IBSSC), pp. 1-4. IEEE, 2021.

[16] Pachhala, Nagababu, S. Jothilakshmi, and Bhanu Prakash Battula. "A comprehensive survey on identification of malware types and malware classification using machine learning techniques." In 2021 2nd international conference on smart electronics and communication (ICOSEC), pp. 1207-1214. IEEE, 2021.

[17] Divakarla, Usha, K. Hemant Kumar Reddy, and K. Chandrasekaran. "A novel approach towards windows malware detection system using deep neural networks." Procedia Computer Science 215 (2022): 148-157.

[18] Hakobyan, Robert, and Timur Jamgharyan. "Polymorphic Malware Analysis Model." In Computer Science and Information Technologies: Proceedings of the 14th International Conference,[Yerevan, Armenia], September, pp. 25-30.

[19] Pompura, Mike. "Improved Detection of Multi-Faceted Polymorphic Malware." Master's thesis, Florida Institute of Technology, 2021.

[20] Damaševičius, Robertas, Algimantas Venčkauskas, Jevgenijus Toldinas, and Šarūnas Grigaliūnas. "Ensemble-based classification using neural networks and machine learning models for windows pe malware detection." Electronics 10, no. 4 (2021): 485.

[21] Alessandro Panconesi, Marian, Will Cukierski, and WWW BIG - Cup Committee. Microsoft Malware Classification Challenge (BIG 2015). https://kaggle.com/competitions/malware-classification, 2015. Kaggle.

[22] Iosifache, A. DikeDataset: A labeled dataset of benign and malicious files. GitHub repository, https://github.com/iosifache/DikeDataset, 2021.

[23] allenye66. Malware Opcodes (final-opcodes). Kaggle, n.d., https://www.kaggle.com/datasets/allenye66/final-opcodes. Accessed 17 Dec. 2025.