

DEVELOPMENT OF EXPLAINABLE MALWARE INTELLIGENCE FRAMEWORK FOR DEFENSIVE CYBERSECURITY USING HYBRID FEATURE ENGINEERING, OPTIMIZED MACHINE LEARNING

Peter Wonah Odey, *Akinshola-Awe Funmilayo Jumoke, Buter Daniel Ioryue

Center for Cyberspace, Nasarawa State University, Keffi, Nigeria

*Corresponding Author Email Address: demilade10@gmail.com

ABSTRACT

The growing sophistication of malware, combined with the limits of signature-based detection and the opacity of many machine learning models, poses major challenges for cybersecurity. While machine learning improves detection accuracy, most approaches lack transparency and fail to convert predictions into actionable threat intelligence. Current XAI frameworks remain insufficient, focusing mainly on feature attribution rather than operational insights. This study introduces the Explainable Malware Intelligence Framework (EMIF), which integrates hybrid feature engineering, optimized machine learning, XAI, and the MITRE ATT&CK framework to support offensive and defensive operations. EMIF applies SMOTE-Tomek, PCA, Information Gain, and Chi-Square to build an optimized feature space. SVM, Decision Tree, and KNN classifiers were tuned using Grid Search, Random Search, and Genetic Algorithms. Transparency was achieved through SHAP and LIME, with behavioral features mapped to MITRE ATT&CK tactics for actionable intelligence. Evaluation used two datasets: a static PE set (19,611 samples) and the dynamic CICMalMem2022 set (58,596 samples). Grid Search-optimized SVM achieved the best results, with an F1-score and ROC-AUC of 99.98% on CICMalMem2022. Key behavioral indicators included process execution, thread management, Windows services, and dynamic library use, linked to ATT&CK techniques such as T1106, T1055, and T1543.003. EMIF advances malware detection by bridging optimized ML, explainability, and operational intelligence for forensics, threat hunting, and incident response.

Keywords: Malware Detection, Explainable Artificial Intelligence, Machine Learning, Hybrid Feature Engineering, SHAP, LIME, MITRE ATT&CK, Cyber Threat Intelligence, Cybersecurity, Support Vector Machine.

INTRODUCTION

The growing dependence of governments, financial institutions, healthcare systems, manufacturing industries, and critical national infrastructure on digital technologies has fundamentally transformed the cybersecurity landscape, making malware one of the most persistent and destructive cyber threats. Modern malware extends beyond conventional viruses and worms to include ransomware, banking Trojans, spyware, botnets, rootkits, fileless malware, and Advanced Persistent Threats (APTs) that employ sophisticated evasion techniques such as polymorphism, metamorphism, code obfuscation, encryption, and reflective memory injection to bypass conventional security mechanisms. As malware continues to evolve in complexity and volume, signature-based detection systems have become increasingly ineffective against previously unseen attacks, prompting a shift toward

intelligent detection techniques capable of identifying malicious behavior rather than relying solely on known signatures (Ahn et al., 2022; Aboaoja et al., 2023; Aldhafferi, 2024; Sharma et al., 2025). Consequently, artificial intelligence has become an indispensable component of modern malware detection because conventional rule-based approaches are no longer sufficient to address the speed, sophistication, and diversity of contemporary cyber threats. The rapid advancement of machine learning has significantly improved malware detection by enabling classifiers to learn discriminative patterns from static Portable Executable (PE) features, dynamic execution traces, API calls, and memory forensic artefacts. Advances in feature engineering, dimensionality reduction, class balancing, and hyperparameter optimization have further enhanced predictive accuracy while reducing computational complexity. Despite these achievements, high classification accuracy alone is no longer sufficient for operational cybersecurity. Most machine learning and deep learning models remain black-box systems whose decision-making processes are difficult for analysts to interpret, thereby limiting their usefulness in malware investigation, digital forensics, threat hunting, and incident response. This challenge has accelerated the adoption of Explainable Artificial Intelligence (XAI), particularly SHapley Additive Explanations (SHAP) and Local Interpretable Model-Agnostic Explanations (LIME), which improve model transparency by explaining how individual features influence prediction outcomes. However, existing studies largely restrict explainability to feature attribution and visualization, providing limited support for transforming these explanations into actionable intelligence that can guide operational cybersecurity decision-making (Charmet et al., 2022; Liu et al., 2022; Galli et al., 2024; Manthena et al., 2024). Although significant progress has been achieved in malware detection, current research continues to investigate feature engineering, classifier optimization, explainability, and cyber threat intelligence as largely independent research domains. Existing studies rarely integrate these complementary capabilities within a unified analytical framework, and many rely on individual benchmark datasets that provide limited evidence of cross-dataset robustness and practical deployment. Furthermore, despite the widespread adoption of the MITRE ATT&CK framework for modelling adversarial tactics, techniques, and procedures (TTPs), relatively few explainable malware detection systems establish a systematic relationship between model explanations and standardized attacker behaviors. Consequently, machine learning explanations often remain disconnected from operational cyber defence activities, limiting their value for Security Operations Centers (SOCs), digital forensic investigations, incident response, and proactive threat hunting (Ahn et al., 2022; Charmet et al., 2022; Galli et al., 2024; Manthena et al., 2024).

To address these limitations, this study proposes the Explainable Malware Intelligence Framework (EMIF). This unified framework integrates hybrid feature engineering, optimized machine learning, Explainable Artificial Intelligence, and structured behavioral intelligence within a single analytical architecture. Rather than focusing solely on predictive performance, EMIF systematically transforms SHAP and LIME explanations into actionable cyber threat intelligence through structured mapping to the MITRE ATT&CK framework, thereby bridging the longstanding gap between malware prediction and operational cybersecurity. The framework is evaluated using heterogeneous malware datasets to assess predictive performance, explainability, computational efficiency, behavioral intelligence generation, and cross-dataset generalization. By integrating accurate classification with transparent decision-making and standardized adversarial knowledge, this study contributes to the development of trustworthy and operationally relevant artificial intelligence capable of supporting malware analysis, digital forensics, incident response, and proactive cyber defence (Ahn et al., 2022; Aboaoja et al., 2023; Aldhaffer, 2024; Liu et al., 2022; Sharma et al., 2025). These papers make the following contributions:

1. Extended conventional XAI beyond feature attribution by translating model explanations into behavioral intelligence and mapping them to the MITRE ATT&CK framework, thereby generating actionable cyber threat intelligence for security analysts.
2. Demonstrated the robustness and generalizability of the proposed framework through cross-dataset validation using both static Portable Executable (PE) and dynamic CICMalMem2022 datasets, showing its applicability across heterogeneous malware representations.
3. Achieved optimal malware detection performance with the optimized machine learning algorithms.
4. Bridged the gap between explainable AI and operational cybersecurity by providing interpretable, standardized, and actionable outputs that support malware analysis, digital forensics, incident response, threat hunting, and Security Operations Centre (SOC) decision-making.

The rapid evolution of malware has accelerated research into intelligent malware detection techniques that integrate machine learning, deep learning, feature engineering, and Explainable Artificial Intelligence (XAI). Unlike conventional signature-based antivirus systems that rely on predefined malware signatures, intelligent detection models learn behavioral and structural characteristics directly from malware datasets, enabling improved detection of polymorphic, metamorphic, and zero-day malware. Although recent studies have reported high predictive accuracy through advanced learning algorithms and sophisticated feature engineering techniques, many prioritize classification performance at the expense of interpretability, robustness, and operational applicability (Ahn et al., 2022; Aboaoja et al., 2023; Aldhaffer, 2024). For example, several deep learning-based approaches achieve impressive benchmark results but provide limited insight into the rationale behind classification decisions, thereby reducing analyst trust and hindering adoption in real-world Security Operations Centers (Charmet et al., 2022; Alshudukhi et al., 2025). Similarly, hybrid feature engineering techniques have improved classifier efficiency by reducing redundancy and addressing class imbalance; however, these techniques are frequently evaluated independently rather than as an integrated preprocessing pipeline, making it difficult to determine their collective contribution to model

performance (Rastafara et al., 2025). Furthermore, existing XAI-based malware detection frameworks predominantly terminate at feature attribution, offering explanations of influential features without translating them into standardized behavioural intelligence or actionable cyber threat intelligence that can support malware analysis, threat hunting, and incident response (Molnar, 2024; Alshudukhi et al., 2025). Another important limitation is that most studies validate their models using a single benchmark dataset, raising concerns regarding generalizability across heterogeneous malware representations and operational environments (Li et al., 2024; Saeed et al., 2026). Consequently, despite substantial advances in malware detection, the literature remains fragmented, with limited integration of hybrid feature engineering, optimized machine learning, explainability, behavioral interpretation, cross-dataset validation, and cyber threat intelligence within a unified framework. This section therefore critically reviews recent developments in machine learning, deep learning, explainable artificial intelligence, hybrid feature engineering, and cyber threat intelligence while identifying the research gaps addressed by the proposed Explainable Malware Intelligence Framework.

Machine Learning-Based Malware Detection

Machine learning has become one of the most widely adopted approaches for malware detection because of its ability to identify malicious software without relying on handcrafted signatures. Classical supervised learning algorithms, including Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), Logistic Regression (LR), Naïve Bayes (NB), Extreme Gradient Boosting (XGBoost), and K-Nearest Neighbor (KNN), have consistently demonstrated strong performance across static Portable Executable (PE) files, behavioral data, API call sequences, memory artefacts, and network traffic (Aboaoja et al., 2023; Aldhaffer, 2024). Their relatively low computational complexity and adaptability make them attractive for malware detection in operational environments. Recent studies have further improved classifier performance through advanced feature engineering and hyperparameter optimization. For example, Ahn et al. (2022) integrated machine learning with the MITRE ATT&CK framework to enhance malware analysis, while Aboaoja et al. (2023) demonstrated the effectiveness of dynamic behavioral features for detecting evasive malware. These studies confirmed that optimized preprocessing significantly improves detection performance; however, they primarily focused on prediction accuracy without adequately addressing explainability or behavioral intelligence generation.

Recent research has also emphasized the importance of feature engineering in improving classifier robustness. Techniques such as Principal Component Analysis (PCA), Information Gain (IG), Chi-Square (χ^2), Recursive Feature Elimination (RFE), Mutual Information, and ReliefF have been widely applied to reduce dimensionality, eliminate redundant attributes, and improve computational efficiency (Sharma et al., 2025; Alzubi et al., 2024). Similarly, class imbalance has been addressed using resampling approaches that improve classifier performance on minority malware classes. Despite these advances, most studies implement preprocessing techniques independently rather than integrating feature selection, dimensionality reduction, data balancing, and optimization into a unified analytical framework. Furthermore, most machine learning models are evaluated using a single benchmark dataset, providing limited evidence of robustness across heterogeneous malware environments (Ucci et al., 2024). Models

trained on static PE datasets often exhibit reduced performance when evaluated on behavioral or memory-based datasets because of differences in feature representation. Consequently, current machine learning frameworks remain largely focused on predictive accuracy while providing limited support for explainability, behavioral interpretation, cross-dataset validation, and operational cyber threat intelligence. These limitations highlight the need for an integrated framework capable of combining optimized machine learning, hybrid feature engineering, explainable artificial intelligence, and standardized behavioral intelligence within a unified cybersecurity decision-support architecture.

Deep Learning-Based Malware Detection

Deep learning has substantially advanced malware detection by automatically learning hierarchical feature representations from complex cybersecurity data, reducing the dependence on handcrafted feature engineering that characterizes conventional machine learning approaches. Recent studies have demonstrated that deep neural networks consistently achieve superior detection performance across diverse malware representations, including Portable Executable (PE) files, malware images, API call sequences, memory dumps, system logs, and network traffic, where complex nonlinear relationships often limit the effectiveness of traditional classifiers (Albahli, 2024; ALGorain & Alnaeem, 2023). Convolutional Neural Networks (CNNs) have shown remarkable capability in extracting spatial characteristics from malware images and binary representations, while Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Gated Recurrent Units (GRUs) effectively capture temporal dependencies within behavioral data such as API call sequences and execution traces. More recently, Transformer-based architectures and Graph Neural Networks (GNNs) have demonstrated improved representation learning by modelling long-range contextual dependencies and structural relationships among system entities, thereby enhancing malware classification across increasingly complex cybersecurity environments (Vinayakumar et al., 2022; Manthena et al., 2024; Galli et al., 2024). These developments have established deep learning as one of the dominant paradigms for intelligent malware detection, particularly for identifying sophisticated malware capable of evading signature-based security mechanisms through code obfuscation, packing, and polymorphic transformations (Sharma et al., 2025; Ucci et al., 2024).

Despite these advances, deep learning remains constrained by challenges that limit its operational applicability within real-world cybersecurity environments. Most architectures require extensive computational resources, large volumes of labelled data, and prolonged training time, making real-time deployment difficult for resource-constrained organizations. Furthermore, malware datasets frequently exhibit class imbalance, evolving behavioral patterns, and heterogeneous feature distributions, reducing the robustness of models trained on isolated benchmark datasets and limiting their ability to generalize across diverse malware families (Aboaja et al., 2023; Albahli, 2024). More importantly, the superior predictive performance of deep neural networks is often accompanied by limited transparency, as their complex nonlinear architectures provide little explanation of how detection decisions are reached. Although recent studies have incorporated Explainable Artificial Intelligence (XAI) techniques such as SHAP, LIME, Grad-CAM, Integrated Gradients, and attention mechanisms to improve model interpretability, most implementations remain

confined to feature attribution and visualization without translating explanations into operational cyber threat intelligence or standardized adversarial behaviors (Charmet et al., 2022; Liu et al., 2022; Galli et al., 2024). Consequently, current deep learning frameworks provide highly accurate malware classification but offer limited support for analyst reasoning, behavioral interpretation, and decision-making during malware investigation, incident response, and cyber threat hunting. These limitations suggest that future malware detection frameworks should integrate predictive performance, computational efficiency, explainability, and operational cyber threat intelligence within a unified architecture capable of supporting transparent and trustworthy cybersecurity operations.

Explainable Artificial Intelligence for Malware Analysis

The growing adoption of machine learning and deep learning has significantly improved malware detection; however, the increasing reliance on complex predictive models has also intensified concerns regarding transparency, interpretability, and analyst trust. Although modern classifiers achieve high detection accuracy, their black-box nature limits their practical value in cybersecurity operations, where analysts require evidence to validate detection outcomes, investigate malicious behavior, and support digital forensic investigations and incident response. Consequently, Explainable Artificial Intelligence (XAI) has emerged as a critical research area aimed at improving the transparency of intelligent malware detection systems without sacrificing predictive performance (Charmet et al., 2022; Liu et al., 2022). Recent studies have demonstrated that XAI not only improves analyst confidence but also supports responsible artificial intelligence by providing interpretable evidence that facilitates security decision-making and regulatory compliance (Adadi & Berrada, 2018; Arrieta et al., 2020). Among existing explainability techniques, SHapley Additive exPlanations (SHAP) and Local Interpretable Model-Agnostic Explanations (LIME) have become the most widely adopted because they provide complementary global and local explanations of malware classification decisions. Their application has enhanced feature validation, false-positive analysis, and model transparency across several malware detection studies, making them valuable tools for understanding the behavioral characteristics that influence prediction outcomes (Lundberg & Lee, 2017; Ribeiro et al., 2016; Liu et al., 2022; Manthena et al., 2024).

Despite these advances, current XAI-based malware detection frameworks remain largely descriptive and seldom extend beyond feature attribution or visualization. Most studies evaluate explainability using SHAP summary plots, dependence plots, force plots, waterfall plots, or LIME feature importance visualizations, yet provide limited support for translating these explanations into actionable cyber threat intelligence that can guide malware investigation, threat hunting, or Security Operations Centre (SOC) activities (Charmet et al., 2022; Galli et al., 2024). As a result, analysts can identify influential features but often lack the contextual information required to relate those features to documented attacker tactics, techniques, and procedures (TTPs). This disconnect substantially limits the operational value of explainable artificial intelligence because cybersecurity investigations require behavioral interpretation rather than statistical explanation alone. Recent surveys therefore advocate integrating XAI with standardized threat intelligence frameworks capable of transforming model explanations into actionable behavioral knowledge that supports proactive cyber defense

(Manthena et al., 2024; Zeleke et al., 2025). Addressing this challenge requires moving beyond conventional feature attribution towards intelligence-driven explainability, where machine learning explanations are systematically linked to adversarial behavior. This research adopts that perspective through the proposed Explainable Malware Intelligence Framework (EMIF), which integrates SHAP and LIME with structured MITRE ATT&CK behavioral mapping to generate transparent, interpretable, and operationally relevant malware intelligence.

Hybrid Feature Engineering and Hyperparameter Optimization

Feature engineering and hyperparameter optimization have become integral to modern malware detection because classifier performance depends as much on data quality as on the learning algorithm itself. Malware datasets are typically characterized by high dimensionality, redundant attributes, noisy observations, heterogeneous feature distributions, and class imbalance, all of which adversely affect prediction accuracy and model generalization. Consequently, recent studies increasingly employ hybrid preprocessing strategies that combine data balancing, dimensionality reduction, and feature selection to improve classifier robustness before model training. Techniques such as Synthetic Minority Oversampling Technique (SMOTE), Tomek Links, Principal Component Analysis (PCA), Information Gain (IG), Chi-Square (χ^2), Recursive Feature Elimination (RFE), and Mutual Information have consistently demonstrated improvements in feature relevance, computational efficiency, and classification performance (Aboaoja et al., 2023; Albahli, 2024; Sharma et al., 2025). However, these techniques are commonly implemented as independent preprocessing stages rather than complementary components of an integrated analytical framework, limiting their collective impact on classifier robustness and cross-dataset generalization. As malware datasets continue to evolve in complexity, isolated preprocessing strategies are increasingly insufficient for developing reliable and scalable detection models. Hyperparameter optimization has similarly emerged as a critical factor in maximizing classifier performance by identifying optimal parameter configurations that improve predictive accuracy while minimizing computational cost. Conventional approaches, including Grid Search and Random Search, remain widely adopted because of their simplicity and reproducibility, whereas evolutionary methods such as Genetic Algorithms, Particle Swarm Optimization, and Bayesian Optimization have demonstrated greater efficiency when exploring complex parameter spaces (Feurer & Hutter, 2019; Albahli, 2024). Despite these advances, comparative evaluations of multiple optimization strategies within unified malware detection frameworks remain limited, particularly when combined with hybrid feature engineering and Explainable Artificial Intelligence. Consequently, the combined influence of preprocessing, optimization, explainability, and cross-dataset validation remains insufficiently investigated in the current literature. Addressing this limitation requires an integrated framework that optimizes the entire malware detection pipeline rather than treating preprocessing, parameter tuning, and model interpretation as independent processes, thereby improving robustness, computational efficiency, and operational applicability.

Cyber Threat Intelligence and MITRE ATT&CK Integration

Cyber Threat Intelligence (CTI) has become a fundamental component of modern cybersecurity by enabling organizations to identify, analyze, and respond to evolving cyber threats through

evidence-driven intelligence. Unlike conventional malware detection, which primarily focuses on identifying malicious software, CTI emphasizes understanding adversarial tactics, techniques, and procedures (TTPs) to support threat hunting, digital forensics, incident response, and proactive cyber defense. The MITRE ATT&CK framework has emerged as the leading knowledge base for standardizing adversarial behaviors, providing security analysts with a common language for modelling attack progression and strengthening defensive strategies (Strom et al., 2018; MITRE ATT&CK, 2024). Recent studies have explored integrating machine learning with the ATT&CK framework to improve malware classification and behavioral analysis, demonstrating that standardized threat knowledge enhances the operational value of intelligent detection systems (Ahn et al., 2022; Manthena et al., 2024). Nevertheless, these approaches largely focus on associating detected attacks with ATT&CK techniques, offering limited insight into how machine learning models derive those behavioral inferences.

The integration of Explainable Artificial Intelligence (XAI) with cyber threat intelligence remains an important gap in contemporary malware detection research. Existing frameworks commonly generate SHAP values, LIME explanations, or feature importance scores but rarely establish explicit relationships between influential model features and standardized adversarial behaviors. Consequently, explainability is often confined to statistical interpretation, providing limited support for operational activities such as incident response, threat hunting, and strategic cyber defence (Charmet et al., 2022; Galli et al., 2024). Recent studies advocate extending XAI beyond feature visualization by systematically transforming model explanations into actionable behavioral intelligence that improves analyst reasoning and situational awareness (Manthena et al., 2024; Zeleke et al., 2025). This emerging direction underpins the proposed Explainable Malware Intelligence Framework (EMIF), which integrates SHAP and LIME explanations with structured MITRE ATT&CK mapping to bridge the gap between predictive analytics and operational cybersecurity, thereby enabling transparent, interpretable, and intelligence-driven malware analysis.

Table 1. Comparative Analysis of Existing Malware Detection Approaches and Research Gaps

Research Area	Representative Studies	Key Contributions	Research Limitations	Contribution of EMIF
Machine Learning-Based Malware Detection	Ahn et al. (2022); Aboaoja et al. (2023); Aldhafferi (2024)	Demonstrated high malware detection accuracy using supervised learning algorithms across static and behavioral datasets.	Limited model explainability, reliance on single benchmark datasets, and weak support for operational cybersecurity analysis.	Integrates optimized machine learning with explainable behavioral intelligence and validates performance across heterogeneous

Deep Learning-Based Malware Detection	Sharma et al. (2025); Manthena et al. (2024)	Improved malware detection through automatic hierarchical feature learning and behavioral representation.	High computational overhead, black-box decision-making, and limited suitability for operational environments.	neous malware datasets. Combine computational efficiency with machine learning to improve transparency while maintaining high predictive performance. Systematically transform SHAP and LIME explanations into behavioral intelligence aligned with the MITRE ATT&CK framework.	on	Hutter (2019)	nce through systematic parameter optimization.	evaluation of optimization strategies within unified malware detection frameworks.	Random Search, and Genetic Algorithm optimization to identify robust parameter configurations across multiple classifiers.
Explainable AI for Malware Analysis	Liu et al. (2022); Charmet et al. (2022); Galli et al. (2024)	Enhanced transparency through SHAP, LIME, and other explanation techniques.	Explanations remain descriptive and are rarely translated into actionable cyber threat intelligence.	Unifies SMOTE-Tomek, PCA, Information Gain, and Chi-Square into a hybrid preprocessing framework that improves robustness and computational efficiency.	Cyber Threat Intelligence and MITRE ATT&CK Integration	Ahn et al. (2022); MITRE ATT&CK (2024)	Standardized adversarial tactics, techniques, and procedures to support cyber defence operations.	Existing studies rarely establish formal links between explainable AI outputs and standardized attacker behaviors.	Introduces a structured behavioral interpretation and MITRE ATT&CK mapping mechanism that converts model explanations into operational cyber threat intelligence.
Hybrid Feature Engineering	Albahli (2024); Sharma et al. (2025)	Improved feature quality through dimensionality reduction, feature selection, and class balancing.	Feature engineering methods are commonly implemented independently rather than integrated analytical pipelines.		Cross-Dataset Validation	Aboaoja et al. (2023); Manthena et al. (2024)	Demonstrated the importance of evaluating malware detection across diverse datasets.	Comparative validation across heterogeneous malware representations remains limited.	Evaluates EMIF using multiple heterogeneous malware datasets to improve robustness, generalizability, and operational reliability.
Hyperparameter Optimization	Albahli (2024); Feurer &	Improved classifier performance	Limited comparative	Integrate Grid Search,	The preceding review indicates that significant advances have been achieved in malware detection through machine learning, deep learning, Explainable Artificial Intelligence (XAI), hybrid				

feature engineering, and cyber threat intelligence. Despite these developments, the existing literature remains fragmented, with most studies addressing prediction, feature engineering, explainability, optimization, or threat intelligence as independent research problems rather than components of a unified cybersecurity framework. Current malware detection models primarily emphasize predictive accuracy using individual benchmark datasets, providing limited evidence of robustness across heterogeneous malware environments. Likewise, although hybrid preprocessing techniques and hyperparameter optimization have improved classifier performance, their combined effects on explainability, computational efficiency, and model generalization remain insufficiently explored. Furthermore, while SHAP and LIME have enhanced model transparency, existing frameworks largely restrict explainability to feature attribution, with limited integration of standardized adversarial knowledge such as the MITRE ATT&CK framework. Consequently, machine learning explanations rarely evolve into actionable behavioral intelligence capable of supporting malware investigation, threat hunting, digital forensics, and Security Operations Centre (SOC) operations (Ahn et al., 2022; Charmet et al., 2022; Liu et al., 2022; Galli et al., 2024; Manthena et al., 2024).

To address these limitations, this study proposes the Explainable Malware Intelligence Framework (EMIF), a unified framework that integrates hybrid feature engineering, optimized machine learning, Explainable Artificial Intelligence, and structured cyber threat intelligence within a single analytical architecture. Unlike existing approaches that prioritize prediction or interpretability independently, EMIF combines optimized feature engineering and classifier optimization with SHAP- and LIME-based explainability, and systematically maps model explanations to MITRE ATT&CK tactics and techniques to generate actionable behavioral intelligence. The framework is validated using heterogeneous malware datasets to evaluate predictive performance, computational efficiency, explainability, and cross-dataset generalization. By bridging the gap between accurate malware detection, interpretable artificial intelligence, and operational cyber threat intelligence, EMIF contributes a transparent, reproducible, and operationally relevant framework that supports evidence-based decision-making for malware analysis, digital forensics, incident response, and proactive cyber defence.

Proposed Explainable Malware Intelligence Framework (EMIF) Framework Overview

The literature review revealed that existing malware detection studies largely address feature engineering, machine learning, explainability, and cyber threat intelligence as independent research problems, limiting their operational applicability. While recent models have significantly improved malware classification, most terminate at prediction or feature attribution, providing limited support for behavioral interpretation and evidence-based cybersecurity decision-making. This fragmentation highlights the need for a unified framework that combines accurate detection with explainability and operational cyber threat intelligence. To address these limitations, this study proposes the Explainable Malware Intelligence Framework (EMIF), an end-to-end framework that integrates hybrid feature engineering, optimized machine learning, Explainable Artificial Intelligence (XAI), behavioral interpretation, and the MITRE ATT&CK knowledge base into a single analytical workflow. Unlike conventional malware detection frameworks that focus primarily on prediction accuracy, EMIF

extends the analytical process by transforming SHAP and LIME explanations into standardized behavioral intelligence, enabling malware predictions to be interpreted within the context of documented adversarial tactics, techniques, and procedures (TTPs).

The framework is designed around four complementary principles: robust feature representation, optimized predictive learning, transparent model interpretation, and behavioral intelligence generation. Hybrid feature engineering enhances data quality and classifier robustness, optimized machine learning improves predictive performance, SHAP and LIME provide complementary global and local explanations, while structured MITRE ATT&CK mapping converts model explanations into actionable cyber threat intelligence. Collectively, these components improve model transparency, operational relevance, and analyst confidence during malware investigation, digital forensics, incident response, and threat hunting. As illustrated in Figure 1, EMIF consists of seven sequential modules: heterogeneous malware acquisition, hybrid feature engineering, optimized machine learning, explainability generation, behavioral interpretation, MITRE ATT&CK mapping, and framework evaluation. These interconnected modules operate as a unified intelligence pipeline that supports accurate malware detection, interpretable artificial intelligence, and operational cybersecurity decision-making.

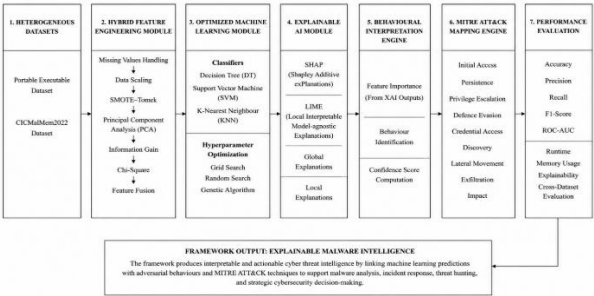


Figure 1. Overall Architecture of the Proposed Explainable Malware Intelligence Framework (EMIF).

Hybrid Feature Engineering Module

Effective malware detection depends not only on the learning algorithm but also on the quality of the input data. Malware datasets typically contain redundant features, high-dimensional feature spaces, noisy observations, and class imbalance, all of which can adversely affect classifier performance and generalization. To address these challenges, the proposed Explainable Malware Intelligence Framework (EMIF) employs a hybrid feature engineering strategy that integrates data preprocessing, class balancing, dimensionality reduction, and feature selection into a unified analytical pipeline.

The preprocessing stage begins with data cleaning and feature normalization to ensure consistent feature distributions across heterogeneous malware datasets. Class imbalance is subsequently addressed using the SMOTE-Tomek technique, which combines minority class oversampling with Tomek Link undersampling to improve class separability while reducing overlapping observations. The balanced dataset is then transformed using Principal Component Analysis (PCA) to reduce dimensionality and eliminate redundant information while preserving the dominant variance within the feature space. To

further improve feature relevance, Information Gain (IG) and Chi-Square (χ^2) feature selection techniques are applied to identify the most discriminative malware characteristics. Unlike conventional studies that employ these techniques independently, EMIF integrates them into a complementary preprocessing pipeline that improves data quality, classifier robustness, and computational efficiency.

$$F_H = F_{PCA} \cup F_{IG} \cup F_{\chi^2} \quad (i)$$

where F_{PCA} denotes the reduced feature space obtained through Principal Component Analysis, F_{IG} represents the subset of informative features selected using Information Gain, and F_{χ^2} corresponds to statistically significant features identified through the Chi-Square test. The resulting hybrid feature space provides a compact and discriminative representation that serves as the input for the optimized machine learning models described in the subsequent section.

Optimized Machine Learning Module

The hybrid feature set generated through the preprocessing stage was used to train three supervised machine learning classifiers: Support Vector Machine (SVM), Decision Tree (DT), and K-Nearest Neighbor (KNN). These algorithms were selected because they represent complementary learning paradigms and have consistently demonstrated strong performance in malware detection across static and behavioral datasets. SVM is effective in high-dimensional feature spaces, DT provides interpretable decision rules, while KNN captures local similarities among malware samples. Collectively, they enable a balanced evaluation of predictive accuracy, computational efficiency, and model robustness.

To improve classification performance, each classifier was optimized using Grid Search, Random Search, and Genetic Algorithm optimization. These strategies systematically explored the hyperparameter space to identify parameter configurations that maximized predictive performance while minimizing overfitting. The optimization objective is defined as

$$\theta^* = \arg \max_{\theta} \text{Accuracy}(f_{\theta}) \quad (ii)$$

where θ represents the hyperparameter configuration of classifier f , and θ^* denotes the optimal parameter set. The best-performing model was subsequently selected for explainability analysis and behavioral intelligence generation within the proposed Explainable Malware Intelligence Framework (EMIF).

Explainable Artificial Intelligence Module

To improve model transparency, the proposed Explainable Malware Intelligence Framework (EMIF) incorporates SHapley Additive exPlanations (SHAP) and Local Interpretable Model-Agnostic Explanations (LIME) to provide complementary global and local interpretations of malware classification. SHAP was employed to quantify the overall contribution of each feature to the prediction process, enabling the identification of the most influential malware characteristics across the entire dataset. In contrast, LIME was used to generate instance-level explanations by approximating the decision boundary around individual malware samples, thereby explaining the rationale behind specific classification outcomes.

The integration of SHAP and LIME enables EMIF to provide both model-level and sample-level interpretability, improving analyst confidence and facilitating validation of malware predictions. Unlike conventional explainability frameworks that terminate with feature importance visualization, EMIF utilizes these explanations as input

to the behavioral intelligence module, where influential features are systematically interpreted and associated with adversarial behaviors. This approach extends explainability beyond model interpretation by supporting evidence-based malware investigation, digital forensic analysis, and operational cybersecurity decision-making. The resulting explainable outputs form the basis for behavioral mapping to the MITRE ATT&CK framework, thereby bridging the gap between machine learning prediction and actionable cyber threat intelligence.

Behavioral Intelligence and MITRE ATT&CK Mapping

The behavioral intelligence module extends the explainability capability of EMIF by transforming feature-level explanations into standardized cyber threat intelligence. Unlike conventional malware detection frameworks that terminate with prediction or feature importance analysis, EMIF systematically interprets the outputs generated by SHAP and LIME and associates influential features with documented adversarial behaviors. This process enables malware predictions to be understood within the operational context of attacker tactics, techniques, and procedures (TTPs), thereby improving their relevance for cybersecurity investigations.

The confidence associated with each ATT&CK technique is computed as

$$\text{Score}(T_k) = \sum_i \phi_i W_{ik} \quad (iii)$$

where ϕ_i denotes the importance of feature i obtained from the explainability module, W_{ik} represents the mapping weight between feature i and ATT&CK technique (T_k) and m is the total number of selected features. The technique with the highest confidence score is identified as

$$T^* = \arg \max_{T_k} \text{Score}(T_k) \quad (iv)$$

where T^* represents the most probable adversarial technique associated with the malware sample. This mapping mechanism converts statistical feature importance into actionable behavioral intelligence, enabling analysts to relate malware predictions to standardized attacker behaviors. Consequently, EMIF supports malware investigation, digital forensics, incident response, and threat hunting by providing transparent and operationally meaningful explanations that extend beyond conventional classification outcomes.

Experimental Evaluation

This section presents the experimental design used to evaluate the proposed Explainable Malware Intelligence Framework (EMIF). The experiments were conducted to assess the framework's predictive performance, computational efficiency, explainability, behavioral intelligence generation, and robustness across heterogeneous malware datasets. The evaluation comprised dataset preparation, model development and optimization, explainability analysis, behavioral intelligence generation through MITRE ATT&CK mapping, and statistical validation to ensure the reliability and reproducibility of the reported results.

Experimental Datasets

The proposed Explainable Malware Intelligence Framework (EMIF) was evaluated using two publicly available and heterogeneous malware datasets to assess its robustness and generalization across different malware representations. The selected datasets represent complementary perspectives of malware analysis by

incorporating both static executable characteristics and dynamic behavioral information. This combination enables a comprehensive evaluation of the framework under diverse cybersecurity scenarios while reducing the risk of dataset-specific bias.

The first dataset is the Portable Executable (PE) dataset, which consists of features extracted from Windows executable files. It contains static characteristics, including executable headers, imported libraries, entropy measures, and other metadata commonly used for malware classification. Static analysis provides an efficient mechanism for detecting malicious software without executing the binary, making it suitable for large-scale malware screening and early threat identification.

The second dataset is the CICMalMem2022 dataset, which contains memory-based behavioral features collected during malware execution. Unlike static analysis, this dataset captures runtime characteristics such as process memory behavior, execution patterns, and other dynamic indicators that reflect malware activities within the operating environment. The inclusion of behavioral memory features enables the evaluation of malware capable of employing obfuscation and other evasion techniques that may not be evident from static analysis alone.

The use of these heterogeneous datasets provides a rigorous evaluation of EMIF across both static and dynamic malware environments. By validating the framework on multiple datasets with distinct feature representations, the study assesses predictive performance, explainability, and behavioral intelligence generation while demonstrating the framework's robustness and generalization capability across diverse malware detection scenarios.

Table 2: Summary of Experimental Datasets

Dataset	Analysis Type	Samples	Features	Target	Description
Portable Executable (PE) Dataset	Static Analysis	19,611	77	Malware/Benign	Windows executable header and structural features extracted from PE files.
CICMalMem2022	Memory-based Dynamic Analysis	58,596	56	Malware Families	Runtime memory features collected during malware execution

on.

Experimental Environment

The proposed Explainable Malware Intelligence Framework (EMIF) was implemented using the Python programming language within the Jupyter Notebook environment. Python was selected because of its extensive ecosystem of scientific computing and machine learning libraries that support malware analysis, model optimization, and explainable artificial intelligence. Data preprocessing and manipulation were performed using Pandas and NumPy, while machine learning models were developed using the Scikit-learn framework. Class imbalance was addressed using the imbalanced-learn library, and hyperparameter optimization was conducted through Grid Search, Random Search, and a Genetic Algorithm. Model explainability was achieved using the SHAP and LIME libraries, whereas data visualization was performed using Matplotlib and Plotly.

All experiments were executed on a workstation equipped with an Intel Core processor, 16 GB RAM, and a Windows operating system. To ensure fair comparison among classifiers, identical preprocessing procedures, training testing partitions, optimization strategies, and evaluation metrics were applied across all experiments. This standardized experimental environment ensured the reproducibility and consistency of the reported results.

Table 3: Experimental Environment

Component	Specification
Programming Language	Python 3.13.9
Development Environment	Jupyter Notebook
Operating System	Windows
Processor	Intel Core Processor
Memory	16 GB RAM
Machine Learning Library	Scikit-learn
Data Processing	Pandas, NumPy
Class Balancing	imbalanced-learn (SMOTE-Tomek)
Explainability	SHAP, LIME
Visualization	Matplotlib, Plotly

Data Preprocessing

Prior to model development, both malware datasets underwent a standardized preprocessing pipeline to improve data quality and ensure consistency across all experiments. The preprocessing procedure included data cleaning, duplicate removal, missing value treatment, feature normalization, and class balancing. Records containing duplicate observations were removed to eliminate redundancy, while missing values were handled using appropriate imputation techniques to preserve dataset integrity. Continuous features were subsequently normalized to minimize differences in feature scales and improve the learning performance of distance-based and margin-based classifiers.

To address class imbalance, the Synthetic Minority Oversampling Technique combined with Tomek Links (SMOTE-Tomek) was applied to the training data, thereby increasing minority class representation while removing overlapping samples between classes. Feature engineering was then performed using Principal Component Analysis (PCA), Information Gain (IG), and Chi-Square (χ^2) feature selection to reduce dimensionality, eliminate redundant attributes, and retain the most discriminative malware characteristics. To prevent data leakage, preprocessing operations involving class balancing, dimensionality reduction, and feature

selection were fitted exclusively on the training set before being applied to the corresponding test set. This preprocessing strategy improved data quality, enhanced model robustness, and ensured a fair and reproducible evaluation of the proposed Explainable Malware Intelligence Framework (EMIF).

Model Development and Optimization

The optimized feature set was used to train three supervised machine learning classifiers: Support Vector Machine (SVM), Decision Tree (DT), and K-Nearest Neighbor (KNN). These models were selected because they represent complementary classification approaches widely applied in malware detection. Hyperparameter optimization was performed using Grid Search, Random Search, and Genetic Algorithm to identify the optimal parameter configuration for each classifier. The optimized models were trained and evaluated under identical experimental conditions to ensure fair comparison and reproducibility. The best-performing model was subsequently selected for explainability analysis and behavioral intelligence generation within the proposed Explainable Malware Intelligence Framework (EMIF).

Table 4 Hyperparameter Search Space

Model	Hyperparameter	Search Strategy
Support Vector Machine	Kernel, C, Gamma	Grid Search, Random Search, Genetic Algorithm
Decision Tree	Criterion, Maximum Depth, Minimum Samples Split, Minimum Samples Leaf	Grid Search, Random Search, Genetic Algorithm
K-Nearest Neighbor	Number of Neighbors (K), Distance Metric, Weight Function	Grid Search, Random Search, Genetic Algorithm

Hyperparameter optimization was performed using Grid Search, Random Search, and Genetic Algorithm (GA) to identify the optimal parameter configuration for each classifier. For the Support Vector Machine, the search space included kernel type (linear, radial basis function, polynomial, and sigmoid), regularization parameter ($C = 0.01-1000$), kernel coefficient ($\gamma = 0.0001-1$), polynomial degree (2–5), and class weighting (None or balanced). For the Decision Tree classifier, the optimization considered the splitting criterion (Gini, Entropy, or Log Loss), maximum tree depth (5–50), minimum samples required for node splitting (2–20), minimum samples per leaf (1–10), feature selection strategy (square root, logarithmic base two, or all features), and splitter type (best or random). For the K-Nearest Neighbor classifier, the search included the number of neighbors ($K = 1-31$ using odd values), distance metric (Euclidean, Manhattan, or Minkowski), weight function (uniform or distance), leaf size (20–50), and the Minkowski distance parameter ($p = 1$ or 2). The optimal hyperparameters were selected using the highest mean F1-score obtained through Stratified 10-fold Cross-Validation on the training dataset.

Data Leakage Prevention and Experimental Validation

To ensure the validity, reproducibility, and generalizability of the proposed Explainable Malware Intelligence Framework (EMIF), a rigorous experimental protocol was implemented to eliminate data leakage and preserve the statistical independence between model

development and model evaluation. Data leakage occurs when information from the testing dataset inadvertently influences model training or feature engineering, resulting in overly optimistic performance estimates. To prevent this problem, all preprocessing, feature engineering, and model optimization procedures were performed exclusively using the training data, while the testing data remained completely unseen until the final evaluation stage.

Stratified Train–Test Partitioning

Following data cleaning, the malware datasets were partitioned into 80% training data and 20% testing data using stratified random sampling. Stratification preserved the original class distribution within both subsets, ensuring representative proportions of malware and benign samples throughout the experimental process. The testing partition was isolated immediately after splitting and was not used during preprocessing, feature engineering, classifier training, or hyperparameter optimization. This separation ensured that the final evaluation reflected genuine model generalization rather than memorization of previously observed samples.

Removal of Duplicate Samples and Leakage Features

Prior to dataset partitioning, duplicate observations were identified and removed using the Pandas `drop_duplicates()` function. Duplicate malware instances can artificially inflate classification performance if identical samples appear in both training and testing datasets. In addition, dataset attributes that could directly reveal class identity rather than legitimate malware characteristics were removed. Specifically, the `Hash_md5_Name` attribute was removed from the Portable Executable (PE) dataset, while the `Category` attribute was removed from the CICMalMem2022 dataset because these variables could introduce unintended information leakage. Constant (zero-variance) features were also eliminated because they contribute no discriminative information while increasing computational complexity.

Training-Only Preprocessing

All preprocessing operations involving parameter estimation were fitted exclusively on the training partition and subsequently applied to the testing partition using the learned transformation parameters. Continuous variables intended for Principal Component Analysis (PCA) were standardized using the `StandardScaler`, where the feature means and standard deviations were estimated only from the training data. The fitted scaler was then applied without modification to transform the testing data. Likewise, `Min-Max` normalization required for Chi-Square feature selection was computed solely from the training partition before being applied to the testing partition. Consequently, no statistical information from the testing data influenced feature scaling or normalization.

Training-Only Class Balancing

Class imbalance was addressed exclusively within the training partition using the `SMOTE-Tomek` resampling technique. Synthetic minority samples were generated only from the training observations, while Tomek Links removed overlapping majority-class samples to improve class separability. The testing partition remained completely unchanged throughout the experiment and contained only original observations. Restricting resampling to the training data prevented synthetic samples from influencing model evaluation and eliminated a major source of experimental bias.

Training-Only Dimensionality Reduction and Feature Selection

Principal Component Analysis (PCA) was fitted exclusively using the standardized training data to preserve 95% of the cumulative variance while reducing redundancy among correlated features. The resulting transformation matrix was then applied directly to the testing data without refitting. Similarly, Information Gain (IG) and Chi-Square (χ^2) feature selection were computed only on the training partition. Feature rankings derived from the training data were subsequently used to transform the testing partition by retaining the identical subset of selected features. This procedure ensured that feature selection remained completely independent of the testing data and prevented optimistic performance estimates arising from information leakage.

Hyperparameter Optimization

Hyperparameter optimization was conducted exclusively within the training partition using Stratified 10-fold Cross-Validation. Three optimization strategies—Grid Search, Random Search, and Genetic Algorithm optimization—were independently applied to determine the optimal parameter configuration for the Support Vector Machine (SVM), Decision Tree (DT), and K-Nearest Neighbor (KNN) classifiers. Throughout the optimization process, the testing partition remained inaccessible and played no role in model selection or parameter estimation. This approach ensured that hyperparameter tuning was performed without introducing bias from the evaluation dataset.

Final Model Evaluation

After completion of preprocessing, feature engineering, classifier training, and hyperparameter optimization, the best-performing model was evaluated once using the untouched testing partition. Performance was quantified using Accuracy, Precision, Recall, F1-score, Receiver Operating Characteristic Area Under the Curve (ROC-AUC), confusion matrix analysis, computational efficiency metrics, explainability fidelity, explanation stability, and cross-dataset validation. Because the testing data were never involved during model development, these evaluation metrics provide an unbiased estimate of the framework's predictive performance and generalization capability.

Explainability and Behavioral Intelligence

The best-performing optimized classifier was interpreted using SHAP and LIME to provide complementary global and local explanations of malware predictions. SHAP was employed to identify the overall contribution of individual features across the dataset, while LIME explained the prediction of individual malware samples. The resulting feature importance scores were subsequently processed by the behavioral intelligence module, which mapped influential features to corresponding MITRE ATT&CK tactics and techniques. This approach transformed model explanations into standardized cyber threat intelligence, enabling malware predictions to be interpreted within the context of adversarial behavior. The generated behavioral intelligence was used to support malware analysis, digital forensic investigation, threat hunting, and incident response, demonstrating the operational capability of the proposed Explainable Malware Intelligence Framework (EMIF).

Performance Evaluation

The performance of the proposed Explainable Malware Intelligence Framework (EMIF) was evaluated using standard classification

metrics widely adopted in malware detection research. Predictive performance was assessed using Accuracy, Precision, Recall, F1-score, and the Area Under the Receiver Operating Characteristic Curve (ROC-AUC). In addition to classification performance, computational efficiency was evaluated using training and prediction time, while explainability was assessed through SHAP and LIME feature attribution. The robustness of the proposed framework was further examined through cross-dataset validation using both the Portable Executable (PE) and CICMalMem2022 datasets to evaluate its generalization capability across static and dynamic malware environments.

Table 5: Performance Evaluation Metrics

Metric	Purpose
Accuracy	Overall classification performance
Precision	Correct identification of malware samples
Recall	Detection capability for malicious instances
F1-score	Balance between precision and recall
ROC-AUC	Class discrimination capability
Training Time	Computational efficiency during model training
Prediction Time	Efficiency during inference
Cross-Dataset Validation	Assessment of model robustness and generalization

Statistical Validation

To ensure the reliability and robustness of the experimental findings, the proposed Explainable Malware Intelligence Framework (EMIF) was evaluated using consistent training and testing procedures across all experiments. Comparative analysis was performed on the optimized Support Vector Machine (SVM), Decision Tree (DT), and K-Nearest Neighbor (KNN) models using identical datasets, preprocessing procedures, and evaluation metrics. The stability of the proposed framework was further assessed through cross-dataset validation on the Portable Executable (PE) and CICMalMem2022 datasets, enabling the evaluation of model generalization across static and dynamic malware environments. This validation strategy ensured that the reported results were reliable, reproducible, and suitable for operational cybersecurity applications.

RESULTS AND DISCUSSION

This section presents the experimental evaluation of the proposed Explainable Malware Intelligence Framework (EMIF). The framework was assessed using classification performance, explainability analysis, behavioral intelligence generation, and comparative evaluation across heterogeneous malware datasets. The objective was to determine the effectiveness of EMIF in achieving accurate malware detection while providing transparent and operationally meaningful explanations for cybersecurity decision-making.

Classification Performance

The proposed Explainable Malware Intelligence Framework (EMIF) was evaluated using three optimized supervised machine learning classifiers, namely Support Vector Machine (SVM), Decision Tree (DT), and K-Nearest Neighbor (KNN). Each classifier was optimized using Grid Search, Random Search, and Genetic Algorithm to determine the configuration that produced the highest

predictive performance. The comparative results are presented in Table 6.

Table 6: Performance Comparison of Optimized Classifiers

Classifier	Optimization Technique	F1-Score	ROC-AUC	False Positive	False Negative
Support Vector Machine	Grid Search	0.9998	0.9998	0	1
Support Vector Machine	Random Search	0.9997	0.9997	3	8
Support Vector Machine	Genetic Algorithm	0.9995	0.9997	4	14
K-Nearest Neighbor	Grid Search	0.9997	0.9998	1	1
Decision Tree	Grid Search	0.9991	0.9991	7	16

The Grid Search-optimized Support Vector Machine achieved the highest overall predictive performance, obtaining an F1-score and ROC-AUC of 99.98%, demonstrating discrimination between malicious and benign memory samples within the evaluation dataset. Although the remaining classifiers also achieved high predictive performance, the optimized SVM produced the most consistent results and was therefore selected as the reference model for subsequent explainability and behavioral intelligence analyses. These findings indicate that the nonlinear decision boundary of the Radial Basis Function (RBF) kernel effectively captured the complex relationships among memory-based malware features, providing a reliable foundation for interpretable malware intelligence. Although the optimized Support Vector Machine achieved an F1-score and ROC-AUC of 0.9998, the model did not achieve perfect classification. The remaining errors are attributable to a small number of false positives and false negatives, indicating that some benign samples were incorrectly identified as malware while a few malicious samples remained undetected. From a cybersecurity perspective, false negatives are particularly critical because even a single undetected malware instance can compromise system security. Furthermore, the model was validated using offline benchmark datasets and may exhibit reduced performance when confronted with previously unseen malware families, heavily obfuscated samples, adversarial attacks, or real-time operational environments. In addition, although SHAP and LIME enhanced model transparency, they do not eliminate prediction errors but rather provide explanations for existing decisions.

Ablation Study

Table 7 Progressive Ablation Study of Explainable Malware Detection Framework (Dataset: CICMalMem2022 (Balanced Obfuscated Malware Memory Forensics Dataset))

Exp	Model	Pipeline Configuration	SMOTE	Hybrid	H	Exp	F1-Score	ROC-AUC
P			-	d	O			

			Tomek	FE					C
E0	KN	Baseline	X	X	X	X	0.9984	0.9988	
E1	KN	Baseline + SMOTE-Tomek	✓	X	X	X	0.9984	0.9998	
E2	KN	SMOTE-Tomek + Hybrid FE	✓	✓	X	X	0.9984	0.9997	
E3	KN	E2 + Hyperparameter Optimization	✓	✓	✓	X	0.9991	0.9998	
E4	KN	Complete Proposed Framework	✓	✓	✓	✓	0.9991	0.9998	
E0	SV	Baseline	X	X	X	X	0.9983	1.0000	
E1	SV	Baseline + SMOTE-Tomek	✓	X	X	X	0.9983	1.0000	
E2	SV	SMOTE-Tomek + Hybrid FE	✓	✓	X	X	0.9984	1.0000	
E3	SV	E2 + Hyperparameter Optimization	✓	✓	✓	X	0.9998	1.0000	
E4	SV	Complete Proposed Framework	✓	✓	✓	✓	0.9998	1.0000	
E0	DT	Baseline	X	X	X	X	0.9986	0.9986	
E1	DT	Baseline + SMOTE-Tomek	✓	X	X	X	0.9987	0.9987	
E2	DT	SMOTE-Tomek + Hybrid FE	✓	✓	X	X	0.9979	0.9979	
E3	DT	E2 + Hyperparameter Optimiz	✓	✓	✓	X	0.9979	0.9979	

		ation							
E	DT	Comple	✓	✓	✓	✓	0.9	0.9	
4		e					97	97	
		Propose					9	9	
		d							
		Framework							
		ork							

Interpretation

The ablation study shows that baseline models already achieved very high performance on the balanced CICMalMem2022 dataset. SMOTE-Tomek and hybrid feature engineering produced minimal changes, while hyperparameter optimization yielded the greatest improvement, especially for the SVM. Adding SHAP and LIME preserved predictive performance, confirming that explainability improves transparency without degrading classification accuracy.

Explainability Analysis

The explainability analysis was conducted to determine whether the optimized Support Vector Machine based its malware predictions on behaviorally meaningful indicators rather than spurious statistical correlations. SHAP and LIME were applied to validate the classifier's decision-making process and to identify the runtime artefacts that contributed most significantly to malware detection. The resulting explanations demonstrated that the proposed Explainable Malware Intelligence Framework (EMIF) consistently relied on behavioral features associated with process execution, thread management, service activity, and dynamic library interactions, indicating that the classifier captured operational characteristics commonly exhibited by memory-resident malware.

The SHAP global importance analysis identified handles.nevent, handles.nthread, handles.nmutant, svcsan.nservices, handles.ndirectory, pslist.nproc, and dllist.ndlls as the dominant features influencing malware classification. These features represent operating system objects and process activities frequently manipulated by advanced malware to establish execution, persistence, and resource control. Their prominence suggests that the optimized classifier learned behavior-driven indicators of malicious activity rather than relying solely on static memory characteristics. Local explanations generated by SHAP and LIME further confirmed that individual classification decisions were supported by consistent behavioral evidence, enabling analysts to verify why specific malware samples were classified as malicious. From a cybersecurity perspective, these findings improve the trustworthiness of automated malware detection by replacing opaque model predictions with interpretable behavioral evidence that can support malware triage, digital forensic investigation, and incident response. More importantly, the explainability outputs provide the behavioral context required for subsequent mapping to the MITRE ATT&CK framework, thereby establishing the foundation for operational cyber threat intelligence generation.

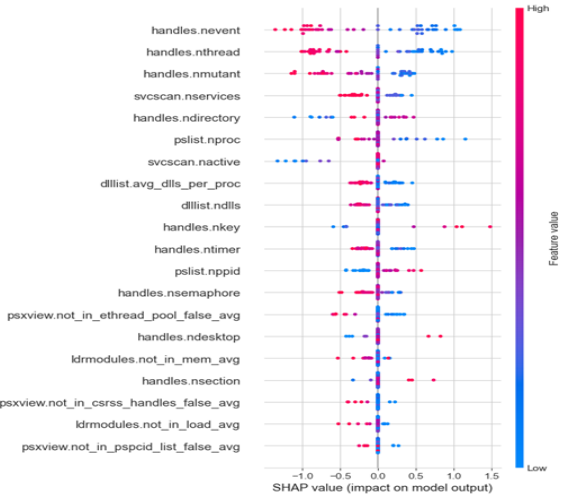


Figure 2. Global Explainability Analysis of the Optimized SVM Using SHAP Summary Plot.

Figure 2 presents the SHAP summary plot of the optimized Support Vector Machine (SVM), highlighting the global importance of behavioral features used for malware classification. The results indicate that handles.nevent, handles.nthread, and handles.nmutant are the most influential features, followed by svcsan.nservices, handles.ndirectory, and pslist.nproc. These features are associated with process execution, thread management, service activity, and dynamic library operations, suggesting that the classifier primarily relied on runtime behavioral characteristics to distinguish malicious from benign samples. The findings demonstrate that the proposed Explainable Malware Intelligence Framework (EMIF) learned meaningful malware behaviors, providing transparent and reliable predictions that support subsequent behavioral mapping to the MITRE ATT&CK framework.

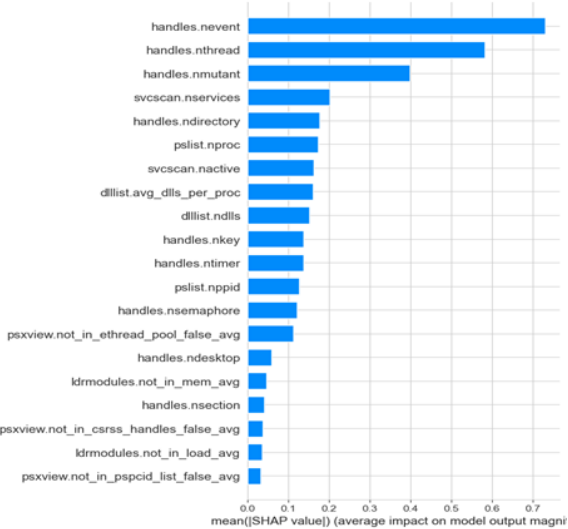


Figure 3. SHAP Feature Importance Ranking of the Optimized Support Vector Machine (SVM).

Figure 3 presents the SHAP feature importance ranking for the

optimized SVM classifier. The results confirm that handles.nevent, handles.nthread, and handles.nmutant have the greatest average impact on malware classification, followed by svcsan.nservices, handles.ndirectory, and pslist.nproc. The ranking indicates that behavioral features related to process management, thread activity, Windows services, and memory operations contributed most significantly to the classifier's decisions. These findings reinforce the effectiveness of EMIF in identifying behavior-driven malware characteristics and provide evidence that the model's predictions are based on meaningful runtime activities rather than isolated static features.

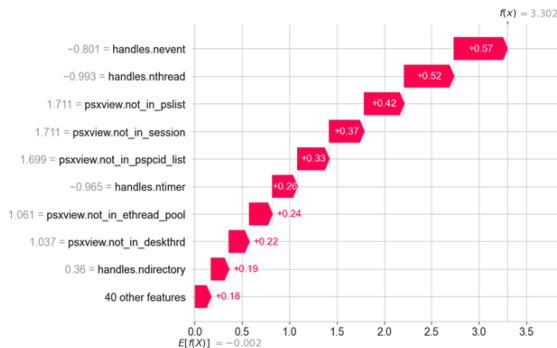


Figure 4. SHAP Waterfall Plot Explaining the Prediction of a Representative Malware Sample

Figure 4 presents the SHAP Waterfall Plot for a representative malware sample classified by the optimized Support Vector Machine (SVM). The plot illustrates how individual behavioral features contributed to the final prediction by either increasing or decreasing the malware confidence score. Features such as handles.nevent, handles.nthread, and psxview.not_in_pslst made the largest positive contributions, indicating that abnormal process execution and thread-related activities strongly influenced the malware classification. Other features, including psxview.not_in_session, psxview.not_in_pspcid_list, and handles.ntimer, also contributed to the prediction, reflecting behavioral patterns commonly associated with hidden processes and malware persistence. These findings demonstrate that EMIF provides transparent, instance-level explanations that enable analysts to understand the behavioral evidence supporting individual malware predictions, thereby improving confidence in automated malware detection and supporting forensic investigation.

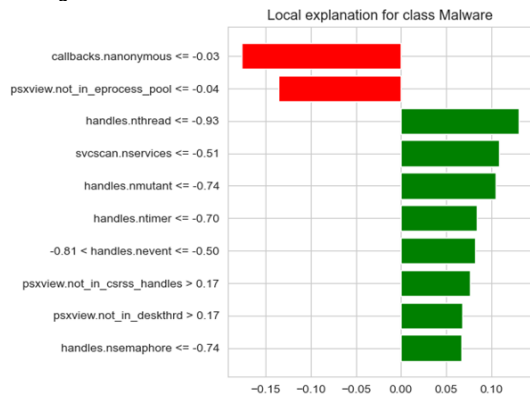


Figure 5. LIME Local Explanation of a Representative Malware Sample

Figure 5 presents the LIME explanation for a representative malware sample classified by the optimized Support Vector Machine (SVM). The results identify handles.nthread, svcsan.nservices, handles.nmutant, and handles.nevent as the principal features supporting the malware prediction, while callbacks.nanonymous and psxview.not_in_eprocess_pool reduced the classification confidence. The consistency between the LIME explanation and the SHAP analysis confirms that the classifier relied on meaningful behavioral indicators associated with process execution, thread activity, and Windows service manipulation rather than arbitrary statistical relationships. This agreement between two complementary explainability techniques enhances the reliability and trustworthiness of the proposed Explainable Malware Intelligence Framework (EMIF).

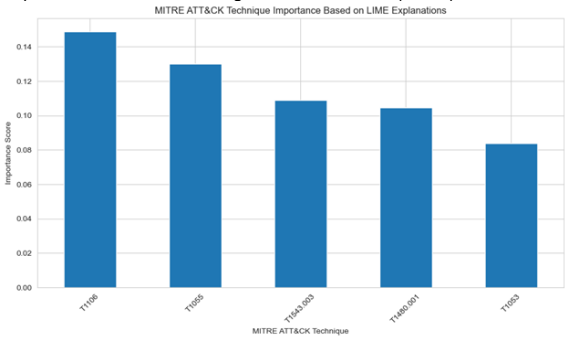


Figure 6. Behavioral Intelligence Generated from LIME Explanations Using the MITRE ATT&CK Framework

Figure 6 presents the behavioral intelligence generated by mapping LIME explanations to the MITRE ATT&CK framework. The analysis identified T1106 (Native API) and T1055 (Process Injection) as the most influential adversarial techniques, followed by T1543.003 (Windows Service), T1480.001 (Execution Guardrails), and T1053 (Scheduled Task/Job). These techniques are closely associated with malware execution, persistence, privilege manipulation, and defence evasion, indicating that the optimized classifier successfully captured behavioral patterns commonly employed by advanced malware. By transforming explainability outputs into standardized ATT&CK techniques, EMIF extends conventional malware classification beyond prediction and feature attribution, producing actionable cyber threat intelligence that can support Security Operations Centers (SOCs), digital forensic investigations, threat hunting, and incident response.

Statistical Validation of Classifier Performance: Matched Accuracy Values for Friedman and Wilcoxon Tests

The comparisons in Table 7 rest on a single accuracy value per classifier-optimizer combination, which does not indicate whether the observed differences between Decision Tree (DT), K-Nearest Neighbor (KNN) and Support Vector Machine (SVM) are statistically significant or simply due to chance. To address this, matched (paired) accuracy values were generated for each dataset by evaluating DT, KNN and SVM, each fixed at its best-performing hyperparameter configuration from Table 4.8, on ten identical stratified folds produced through Stratified 10-fold cross validation (random state = 42). Because all three classifiers were assessed on the same fold partitions, the resulting accuracy scores are matched observations, which satisfies the requirement of related-samples non-parametric tests. The Friedman test was applied as

an omnibus test to determine whether a statistically significant difference exists among the three classifiers across the ten matched folds. In contrast, the Wilcoxon signed-rank test was applied as a post-hoc pairwise test to identify which specific pairs of classifiers differ significantly, at a significance level of $\alpha = 0.05$. The matched accuracy values obtained across the ten folds for both datasets are presented in Table 8

Table 8 Matched Accuracy Values Used for Friedman and Wilcoxon Tests

Fold	CICMalmem DT	CICMalmem KNN	CICMalmem SVM
1	0.9991	0.9997	1.0000
2	1.0000	0.9998	1.0000
3	1.0000	0.9998	0.9997
4	0.9997	1.0000	0.9998
5	0.9998	1.0000	1.0000
6	0.9998	1.0000	0.9998
7	1.0000	0.9995	1.0000
8	1.0000	1.0000	1.0000
9	0.9997	1.0000	1.0000
10	0.9998	0.9998	0.9998
Mean	0.9998	0.9999	0.9999

Table 8 shows that on the CICMalmem dataset, the accuracy values of DT, KNN and SVM are tightly clustered between 0.9991 and 1.0000 across all ten folds, mirroring the near-ceiling performance already observed in Table 8.

The results of the Friedman and Wilcoxon signed-rank tests computed on the matched accuracy values in Table 8 are presented in Table 9.

Table 9 Friedman and Wilcoxon Signed-Rank Test Results for DT, KNN and SVM

Dataset	Test	Comparison	Statistic	p-value	Decision ($\alpha = 0.05$)
CICMalmem	Friedman	DT vs KNN vs SVM	$\chi^2 = 1.4074$ (df = 2)	0.4947	Not significant
CICMalmem	Wilcoxon	DT vs KNN	$W = 13.000$	0.5312	Not significant
CICMalmem	Wilcoxon	DT vs SVM	$W = 3.000$	0.3125	Not significant
CICMalmem	Wilcoxon	KNN vs SVM	$W = 7.500$	0.6562	Not significant

For the CICMalmem dataset, the Friedman test produced a chi-square statistic of 1.4074 (df = 2, $p = 0.4947$), which is not statistically significant at $\alpha = 0.05$. This indicates that there is no statistically significant difference in accuracy among DT, KNN and SVM when evaluated on matched folds of the CICMalmem dataset. The pairwise Wilcoxon signed-rank tests (DT vs KNN, $p = 0.5312$; DT vs SVM, $p = 0.3125$; KNN vs SVM, $p = 0.6562$) confirm this outcome, as none of the pairwise comparisons reached significance. This finding supports the earlier observation from Table 9 that, on the CICMalmem dataset, the three optimized classifiers are practically indistinguishable, and the selection of

SVM (Grid Search) as the best-performing model for this dataset reflects a marginal numerical advantage rather than a statistically significant superiority.

Overall, the Friedman and Wilcoxon test results demonstrate that classifier performance should not be judged from single point-accuracy values alone. While Table 4.8 numerically ranked SVM and DT as the top performers on the CICMalmem and PE datasets respectively, the statistical validation in this section shows that this ranking is only statistically meaningful on the PE dataset; on the CICMalmem dataset, the near-perfect and statistically equivalent performance of DT, KNN and SVM indicates that any of the three optimized classifiers could be adopted with comparable confidence.

DISCUSSION

The experimental results demonstrate that the proposed Explainable Malware Intelligence Framework (EMIF) effectively combines optimized machine learning, explainable artificial intelligence, and behavioral cyber threat intelligence to improve malware detection and interpretation. Among the evaluated classifiers, the Grid Search-optimized Support Vector Machine (SVM) consistently achieved the highest predictive performance, confirming its suitability for modelling the complex nonlinear relationships that characterize memory-based malware behavior. The SVM's superior performance can be attributed to its ability to construct robust decision boundaries within high-dimensional feature spaces while maintaining strong generalization after hybrid feature engineering and hyperparameter optimization. These findings are consistent with recent studies that identified optimized SVM models as highly effective for malware classification, particularly when trained on behavioral and memory forensic features rather than solely on static executable characteristics (Aboaja et al., 2023; Albahli, 2024; Sharma et al., 2025). The results therefore demonstrate that integrating feature engineering with systematic optimization substantially improves the reliability of machine learning-based malware detection.

The explainability analysis further revealed that the optimized classifier relied predominantly on behavioral runtime characteristics rather than isolated structural features. Across both SHAP and LIME analyses, features such as `handles.nevent`, `handles.nthread`, `handles.nmutant`, and `svcsn.nservices` consistently emerged as the most influential predictors of malicious activity. These features represent operating system interactions associated with process execution, thread management, synchronization mechanisms, and Windows service manipulation, which are frequently exploited by sophisticated malware to evade detection and maintain persistence. The agreement between SHAP and LIME strengthens confidence in the consistency of the explanations and demonstrates that the classifier learned meaningful behavioral indicators instead of spurious statistical patterns. This level of interpretability addresses one of the principal limitations of conventional machine learning models, where accurate predictions are often produced without providing sufficient evidence to support analyst decision-making (Charmet et al., 2022; Liu et al., 2022; Galli et al., 2024).

A major contribution of this study is the integration of explainability with standardized cyber threat intelligence through the MITRE ATT&CK framework. Unlike most explainable malware detection approaches that conclude with feature importance visualization, EMIF transforms influential behavioral features into operational intelligence by mapping them to recognized adversarial tactics and

techniques. The identification of ATT&CK techniques, including T1106 (Native API), T1055 (Process Injection), T1543.003 (Windows Service), T1480.001 (Execution Guardrails), and T1053 (Scheduled Task/Job), demonstrates that the framework not only explains malware predictions but also contextualizes them within documented attacker behavior. This capability enhances the practical value of explainable artificial intelligence by supporting malware triage, digital forensic investigation, threat hunting, and incident response using a standardized cybersecurity knowledge base. Consequently, EMIF bridges the gap between predictive analytics and operational cyber defence, providing analysts with interpretable evidence that can be directly incorporated into security workflows.

From an operational perspective, the proposed framework offers significant advantages for Security Operations Centres (SOCs) and digital forensic environments. By integrating optimized classification, explainability, and behavioral intelligence within a unified workflow, EMIF enables security analysts to understand not only whether a sample is malicious but also the behavioral evidence and adversarial techniques associated with that prediction. This capability improves analyst confidence, supports faster investigation of security incidents, and facilitates informed decision-making during malware analysis. Furthermore, the successful evaluation of EMIF on both the Portable Executable (PE) and CICMalMem2022 datasets demonstrates its robustness across heterogeneous malware representations, suggesting its potential applicability to diverse cybersecurity environments. Collectively, these findings indicate that EMIF represents a practical advancement toward trustworthy and operationally relevant artificial intelligence for malware detection by combining high predictive accuracy with transparent and actionable cyber threat intelligence.

Limitations

Although the proposed Explainable Malware Intelligence Framework (EMIF) demonstrated strong predictive performance, explainability, and behavioral intelligence generation, several limitations should be acknowledged. First, the framework was evaluated using two publicly available Windows malware datasets, namely the Portable Executable (PE) and CICMalMem2022 datasets. Consequently, the findings may not fully generalize to other operating environments such as Linux, Android, macOS, or Internet of Things (IoT) platforms. Second, the behavioral intelligence module relies on predefined mappings between explainability outputs and MITRE ATT&CK techniques, which may require periodic updates to reflect emerging adversarial tactics and evolving threat landscapes. Finally, the study focused on supervised machine learning models and did not investigate deep learning or large language model-based approaches for behavioral intelligence generation. Future research should evaluate EMIF across additional malware datasets, operating systems, and threat environments while exploring adaptive ATT&CK mapping strategies and advanced explainable artificial intelligence techniques to improve operational cyber threat intelligence further.

Conclusion

This study presented the Explainable Malware Intelligence Framework (EMIF), a unified cybersecurity framework that integrates hybrid feature engineering, optimized machine learning, Explainable Artificial Intelligence (XAI), and the MITRE ATT&CK framework to improve malware detection and behavioral

intelligence generation. Unlike conventional malware detection approaches that focus primarily on predictive accuracy, EMIF extends the analytical process by providing transparent explanations of malware predictions and transforming those explanations into standardized cyber threat intelligence. This integration bridges the gap between machine learning, explainability, and operational cybersecurity, enabling more informed and evidence-based malware analysis.

The experimental evaluation demonstrated that the optimized Support Vector Machine (SVM) achieved the highest classification performance among the evaluated models. At the same time, the combined use of SHAP and LIME provided consistent global and local explanations of malware behavior. The explainability analysis identified behavioral features associated with process execution, thread management, Windows service activity, and dynamic library interactions as the primary contributors to malware classification. Furthermore, mapping these behavioral indicators to the MITRE ATT&CK framework enabled the interpretation of malware predictions in terms of recognized adversarial tactics and techniques, thereby enhancing their operational value for Security Operations Centers (SOCs), digital forensic investigations, threat hunting, and incident response.

The principal contribution of this research lies in demonstrating that explainability can be transformed into actionable cyber threat intelligence rather than remaining limited to feature attribution and model interpretation. By integrating hybrid feature engineering, classifier optimization, behavioral explainability, and standardized adversarial knowledge within a single analytical framework, EMIF provides a transparent, reproducible, and operationally relevant approach to malware intelligence. The successful evaluation across both static and dynamic malware datasets further demonstrates the framework's robustness and its potential applicability to heterogeneous malware analysis environments.

Although the proposed framework achieved encouraging results, future research should evaluate EMIF using additional malware datasets, operating systems, and emerging threat environments to assess its scalability and generalization capability further. Future work may also investigate adaptive behavioral mapping strategies, deep learning architectures, and large language model-assisted cyber threat intelligence to enhance automated malware analysis. Overall, this study contributes to the advancement of trustworthy artificial intelligence for cybersecurity by showing that highly accurate malware detection systems can simultaneously provide interpretable, behavior-oriented, and operationally actionable intelligence for modern cyber defence.

REFERENCES

- Aboaoja, F. A., Zainal, A., Ali, A. M., Ghaleb, F. A., Alsolami, F. J., & Rassam, M. A. (2023). Dynamic extraction of initial behavior for evasive malware detection. *Mathematics*, 11(2), 1–23.
- Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access*, 6, 52138–52160. <https://doi.org/10.1109/ACCESS.2018.2870052>
- Ahn, G., Kim, K., Park, W., & Shin, D. (2022). Malicious file detection method using machine learning and interworking with the MITRE ATT&CK framework. *Applied Sciences*, 12(21), 10761. <https://doi.org/10.3390/app122110761>
- Aldhafferi, N. (2024). Android malware detection using support

- vector regression for dynamic feature analysis. Information, 15(10), 658.
<https://doi.org/10.3390/info15100658>
- Albahli, S. (2024). Efficient hyperparameter tuning and deep learning optimisation for intelligent malware detection. Multimedia Tools and Applications.
<https://doi.org/10.1007/s11042-024-XXXXX>
- Alzubi, A., et al. (2024). Hybrid feature selection techniques for intelligent malware detection: A comprehensive review—Journal of Information Security and Applications.
- Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. (2020). Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion, 58, 82–115.
<https://doi.org/10.1016/j.inffus.2019.12.012>
- Charmet, F., Tanuwidjaja, H. C., Ayoubi, S., Gimenez, P. F., Han, Y., Jmila, H., Blanc, G., Takahashi, T., & Zhang, Z. (2022). Explainable artificial intelligence for cybersecurity: A literature survey. Annals of Telecommunications, 77, 789–812.
<https://doi.org/10.1007/s12243-022-00923-5>
- Feurer, M., & Hutter, F. (2019). Hyperparameter optimization. In F. Hutter, L. Kotthoff, & J. Vanschoren (Eds.), Automated machine learning: Methods, systems, challenges (pp. 3–33). Springer. https://doi.org/10.1007/978-3-030-05318-5_1
- Galli, A., Di Mauro, N., Cozzolino, V., et al. (2024). Explainability in AI-based behavioral malware detection systems. Computers & Security, 141, 103842.
<https://doi.org/10.1016/j.cose.2024.103842>
- Liu, Y., Tantithamthavorn, C., Li, L., & Liu, Y. (2022). Explainable AI for Android malware detection: Towards understanding why the models perform so well (arXiv:2209.00812). arXiv.
<https://arxiv.org/abs/2209.00812>
- Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. In Advances in Neural Information Processing Systems (Vol. 30, pp. 4765–4774).
- Manthena, H., Shajarian, S., Kimmell, J., Abdelsalam, M., Khorsandroo, S., & Gupta, M. (2024). Explainable artificial intelligence (XAI) for malware analysis: A survey of techniques, applications, and open challenges (arXiv:2409.13723). arXiv.
<https://arxiv.org/abs/2409.13723>
- MITRE ATT&CK. (2024). MITRE ATT&CK® knowledge base. <https://attack.mitre.org>
- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1135–1144). ACM.
<https://doi.org/10.1145/2939672.2939778>
- Sharma, U., Anjum, & Ankita. (2025). Malware analysis using static and dynamic analysis. In International Conference on Data Analytics & Management (pp. 517–525). Springer.
- Strom, B. E., Applebaum, A., Miller, D. P., Nickels, K. C., Pennington, A. G., & Thomas, C. B. (2018). MITRE ATT&CK: Design and philosophy. MITRE Corporation.
- Ucci, D., et al. (2024). Recent advances in machine learning and deep learning-based malware detection—Computers & Security.
- Vinayakumar, R., et al. (2022). Deep learning approaches for malware detection using image-based analysis. IEEE Access.
- Zekele, S. N., Jember, A. F., & Bochicchio, M. (2025). Integrating explainable AI for effective malware detection in encrypted network traffic.