processes

MDPI

Article

MAL-XSEL: Enhancing Industrial Web Malware Detection with an Explainable Stacking Ensemble Model

Ezz El-Din Hemdan [1,2], Samah Alshathri [3,*], Haitham Elwahsh [4,5], Osama A. Ghoneim [6] and Amged Sayed [7,8,*]

1 Structure and Materials Research Lab, Prince Sultan University, Riyadh 11586, Saudi Arabia; ezzeldinhemdan@el-eng.menofia.edu.eg
2 Department of Computer Science and Engineering, Faculty of Electronic Engineering, Menoufia University, Menouf 32952, Egypt
3 Department of Information Technology, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia
4 Faculty of Information Technology, Applied Science Private University, Amman 11931, Jordan; h_elwahsh@asu.edu.jo
5 Department of Computer Science, Faculty of Computers and Information, Kafrelsheikh University, Kafrelsheikh 33511, Egypt
6 Department of Computer Science, Faculty of Computers and Information, Tanta University, Tanta 31527, Egypt; osamaghoneim@ics.tanta.edu.eg
7 Department of Electrical Energy Engineering, College of Engineering & Technology, Arab Academy for Science Technology & Maritime Transport, Smart Village Campus, Giza 12577, Egypt
8 Industrial Electronics and Control Engineering Department, Faculty of Electronic Engineering, Menoufia University, Menouf 32952, Egypt
* Correspondence: sealshathry@pnu.edu.sa (S.A.); amged.sayed@aast.edu (A.S.)

Abstract: The escalating global incidence of malware presents critical cybersecurity threats to manufacturing, automation, and industrial process control systems. Given the fast-developing web applications and IoT devices in use by industry operations, securing a transparent and effective malware detection mechanism has become imperative to operational resilience and data integrity. Classical methods of malware detection are conventionally opaque "black boxes" with limited transparency, thus eroding trust and hindering deployment in security-sensitive contexts. In this respect, this research proposes MAL-XSEL—a malware detection framework using an explainable stacking ensemble learning approach for performing high-accuracy classification and interpretable decision-making. MAL-XSEL explicates the model predictions through Shapley additive explanations (SHAP) and local interpretable model-agnostic explanations (LIME), which enable security analysts to validate how the detection logic works and prioritize the features contributing to the most critical threats. Evaluated on two benchmark datasets, MAL-XSEL outperformed conventional machine learning models, achieving top accuracies of 99.62% (ClaMP dataset) and 99.16% (MalwareDataSet). Notably, it surpassed state-of-the-art algorithms such as LightGBM (99.52%), random forest (99.33%), and decision trees (98.89%) across both datasets while maintaining computational efficiency. A unique interaction of ensemble learning and XAI is employed for detection, not only with improved accuracy but also with interpretable insight into the behavior of malware, thereby allowing trust to be substantiated in an automated system. By closing the divide between performance and interpretability, MAL-XSEL enables cybersecurity practitioners to deploy transparent and auditable defenses against an ever-growing resource of threats. This work demonstrates how there can be no compromise on explainability in security-critical applications and, as such, establishes a roadmap for future research on industrial malware analysis tools.

Academic Editors: Lee Tin Sin, Thomas S.Y. Choong and John Anthony Rossiter

Received: 21 March 2025
Revised: 23 April 2025
Accepted: 24 April 2025
Published: 26 April 2025

Citation: Hemdan, E.E.-D.; Alshathri, S.; Elwahsh, H.; Ghoneim, O.A.; Sayed, A. MAL-XSEL: Enhancing Industrial Web Malware Detection with an Explainable Stacking Ensemble Model. *Processes* **2025**, *13*, 1329. https://doi.org/10.3390/pr13051329

Keywords: malware mitigation; industrial web applications; classification; machine learning; explainable artificial intelligence (XAI); SHAP (Shapley additive explanations); (LIME) local interpretable model-agnostic explanations

1. Introduction

Recently, network security has been under increasing threat due to malware. In the digital era, malicious software, known as malware, presents a significant and evolving threat to security, making its detection a top priority. Malware is designed to compromise the confidentiality, integrity, and functionality of a system [1]. It refers to any malicious software developed to steal data, cause damage, or compromise computer systems. This includes various types, such as viruses, worms, trojans, backdoors, and spyware. Given the ubiquitous use of computers and the Internet in daily life, malware represents a significant threat to information security. Consequently, malware detection remains a critical concern for both the anti-malware industry and researchers.

Figure 1 shows the annual number of malware attacks worldwide, measured in billions, from 2015 to 2023 [2]. This graphical representation provides a comprehensive overview of the evolving landscape of cybersecurity threats over the years. The increasing trend depicted in the graph highlights the alarming rise in the frequency and severity of malware attacks targeting various digital assets and systems globally.

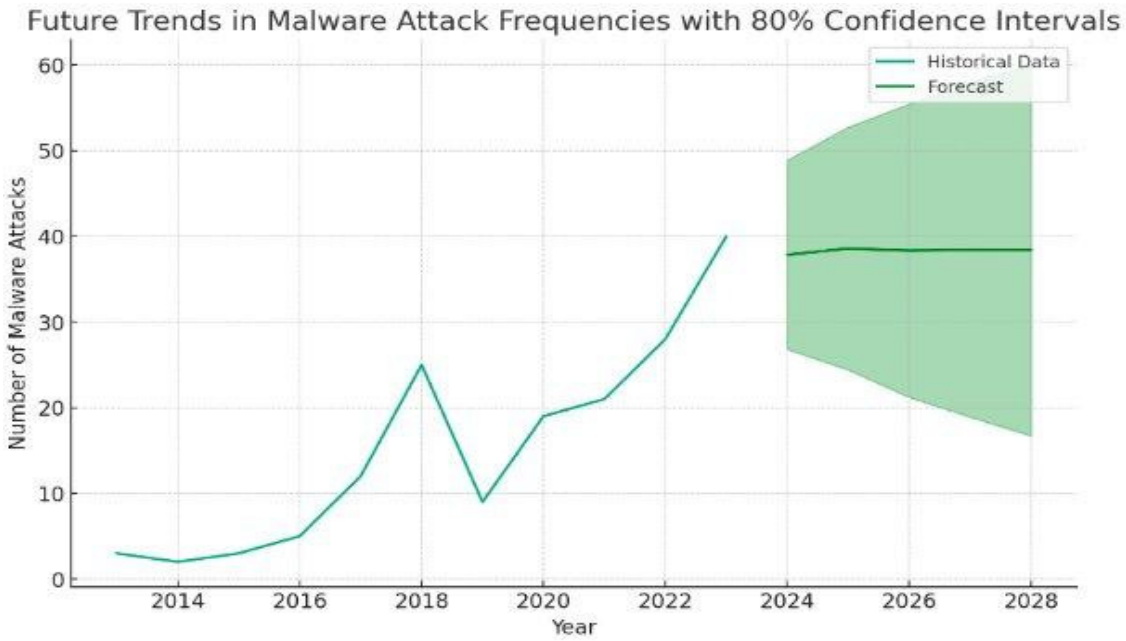


Figure 1. The number of malware attacks worldwide [2].

Each point on the graph represents a specific year, while the corresponding value on the vertical axis denotes the estimated number of malware attacks reported worldwide during that particular year. The continuous upward trajectory underscores the growing sophistication and proliferation of malicious activities perpetrated by cybercriminals across different sectors and industries. Understanding this trend is crucial for cybersecurity professionals, policymakers, and organizations as they strive to develop robust defense mechanisms and proactive strategies to safeguard digital infrastructure and sensitive data from evolving cyber threats. By analyzing historical data and the projections depicted in Figure 1, stakeholders can gain valuable insights into the scale and nature of malware at-

tacks, enabling them to devise effective countermeasures and resilience-building initiatives to mitigate risks and enhance cybersecurity posture in an increasingly digitized world for various applications.

Numerous research efforts have focused on intelligent malware detection through data mining and machine learning techniques. These attempts were successful in many ways yet depended on narrow learning architectures and thus failed to resolve the realities posed by the industrial malware threats. Cloud-based Internet applications are widely used to connect industrial platforms, such as smart manufacturing and automation systems and IoT-enabled production lines. However, widespread connectivity turns out to be a very attractive target for various cyberattacks, including malware that could attack major industrial processes and access sensitive operational data, resulting in financial damage, reputational risk, and even greater destruction in the end. Failure in industrial malware attack cases can result in short supply chains, equipment failure, production downtime, and safety hazards [3–5]. There are many threats to the AI ecosystem that need to be delineated from the security perspective of the proposed model: backdoor attacks, adversarial attacks, and jailbreak attacks. Backdoor attacks subvert training with hidden behavior, while adversarial attacks change the input data to corrupt predictions [6–8]. Jailbreak attacks try to circumvent safety restrictions or constraints imposed on deployed systems, giving rise to threats to real-world industrial applications where security and stability are vital. Therefore, strong and explainable malware detection frameworks should be developed to secure industrial web applications, maintain operational resilience, and keep the Industry 4.0 ecosystem safe against evolving cyber threats. Such frameworks should integrate explainable AI-driven cybersecurity models to detect and mitigate industrial malware threats, ensuring minimal disruption to manufacturing processes, automation systems, and critical infrastructure.

Conversely, machine learning, a branch of artificial intelligence, entails crafting algorithms and statistical models that empower computers to execute tasks without explicit programming. These systems learn from data and derive decisions based on the acquired information. The workflow involves training a model on a dataset, enabling it to discern patterns and render predictions or decisions with novel data. Machine learning finds broad applications, including image recognition, disease predictions, and recommendation systems. Its ability to enhance and evolve renders it a potent instrument for tackling intricate challenges across various domains. In addition, in [9,10], the approach utilizes machine learning for pre-deployment performance prediction in software-defined networks (SDNs), employing neural network boosting regression to enhance accuracy. Additionally, a survey examines the application of graph-based deep learning techniques to communication networks, exploring various models and methodologies that improve network performance and efficiency.

With the increasing prevalence of malware, it has become a critical cybersecurity concern, impacting individual users, corporations, and governments worldwide. This growing threat highlights the urgent need for advanced research in malware detection, particularly for secure web applications. Traditional malware detection models are often treated as “black boxes”, making their decision-making process opaque to users. This lack of transparency reduces trust and limits their adoption in security-critical environments. To address this issue, this study proposes MAL-XSEL, an explainable stacking-based ensemble learning model for web malware classification. By leveraging explainable artificial intelligence (XAI) techniques, this framework enhances interpretability and provides insights into malware detection decisions. XAI methods, such as SHAP (Shapley additive explanations), make machine learning outputs more understandable, allowing security

experts to comprehend the factors influencing predictions. The decision-making workflow of explainable AI (XAI) is shown in Figure 2 [10,11].

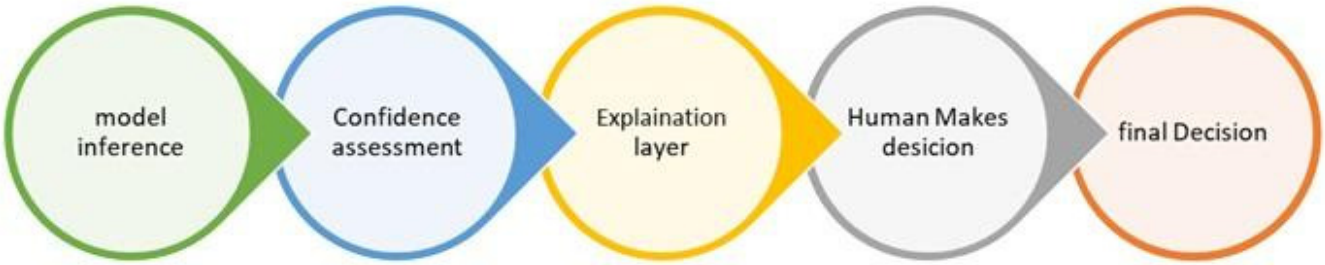


Figure 2. Decision-making workflow of explainable AI (XAI).

This paper presents a framework for efficient malware detection for secure web applications. The proposed framework enhances the efficiency and accuracy of malware detection for securing web applications over the Internet. The results demonstrate that the hybrid framework outperforms traditional machine learning methods across various evaluation metrics. Furthermore, this paper aims to provide an effective malware detection framework to create a robust malware detection system for web applications. The encouraging experimental results inspire the undertaking of large-scale labeled online data collection to assess the proposed framework's generalization capability. The following summarizes the contributions of this paper:

- An efficient and robust framework is developed that effectively secures industrial web applications from malicious activities, using binary classification to specify malware and benign software for malware detection. Furthermore, an experimental analysis of the proposed industrial web-based malware detection framework is performed to classify web traffic accurately and ensure the security of critical manufacturing, automation, and industrial control systems;
- The proposed framework incorporates explainable artificial intelligence (XAI) techniques, such as SHAP and LIME, to enhance the interpretability of malware detection models. The incorporation of these techniques enables security practitioners to know the key features influencing classification decisions, thus closing the gap between model performance and human interpretability;
- The effectiveness of the proposed stacking-based framework is validated through extensive experiments on two benchmark malware datasets. The results demonstrate that the proposed method consistently outperforms conventional machine learning models, confirming its robustness and superior accuracy in industrial malware detection tasks;
- By combining high accuracy with explainability, MAL-XSEL offers a robust and transparent malware detection solution, bridging the gap between performance and interpretability. This framework empowers cybersecurity professionals with better insights into malware classification, making it a practical tool for securing industrial web applications against emerging threats.

The remainder of this paper is structured as follows: Section 2 provides a concise overview of the pertinent literature in this paper's subject area, while Section 3 presents an in-depth explanation of the proposed malware detection framework for web applications. Section 4 delves into the experimental findings and outcomes analysis of the proposed framework. Finally, this paper's conclusion and future scope are presented in Section 5.

2. Previous Studies

Due to their connectivity to external networks, industrial control systems (ICSs) are particularly vulnerable to malware threats. A prominent example is the Stuxnet attack [12], which caused severe disruption to Iran's uranium enrichment facilities, highlighting the devastating impact malware can have on industrial infrastructure. Traditional machine learning-based detection methods have been applied to this domain; however, they often fail to optimize classification performance, especially under real-time and evolving threat conditions. Recent studies have proposed more advanced techniques tailored for industrial settings. For instance, Huang et al. [13] employed opcode2vec feature extraction combined with conditional variational autoencoders (CVAEs) and generative adversarial networks (GANs), achieving high accuracy and robust malware classification within ICS environments. Similarly, Zheng et al. [14] introduced an adaptive fuzzy SIR model for predicting malware propagation in industrial IoT (IIoT) networks. This model dynamically adjusts infection and recovery rates using gradient descent and fuzzy logic, improving its adaptability under uncertain network conditions. Another notable advancement is presented by Naeem et al. [15], who proposed a malware detection method for IIoT devices based on color image visualization of malware binaries, combined with deep convolutional neural networks (CNNs). Their approach demonstrated improved accuracy and faster prediction times compared to conventional methods.

In addition to advancements in ICS-specific malware detection, various studies have demonstrated the effective application of machine learning (ML) techniques in detecting malware across a wide range of file types, including executables, scripts, documents, and images. These models analyze behavioral and structural characteristics to improve detection accuracy, particularly in environments with unsecured networks. For instance, the study in [16] focused on feature selection from portable executable (PE) headers, eliminating non-contributory features to improve the performance of supervised learning models. In [17], the authors proposed a machine learning-based approach for detecting unknown and metamorphic malware by analyzing the frequency of opcode occurrence. Using the Microsoft Malware Classification Challenge dataset and multiple feature selection methods, the study demonstrated that classifiers such as random forest achieved high detection accuracy. Similarly, in [18], the authors introduced a novel cloud-based malware dataset, CMD_2024, which integrates both static and dynamic features to enhance malware detection in cloud environments. By applying synthetic data generation using conditional tabular GANs and leveraging various machine and deep learning classifiers, their approach achieved high detection performance, particularly improving the recognition of rare malware classes.

Furthermore, the authors in [19] proposed a multimodal deep learning framework that fuses convolutional neural network models trained on grayscale images, entropy graphs, and SimHash representations of malware executables. Their work integrates explainability techniques, such as Grad-CAM and t-SNE, and demonstrates state-of-the-art performance even under adversarial attacks, highlighting the potential of robust, interpretable, and feature-rich malware detection models. A notable direction in recent studies is addressing concept drift, the phenomenon where malware evolves over time. The work in [20] emphasized its impact on long-term detection accuracy, suggesting the need for adaptive models. Its findings show that by addressing this, concept drift can be used to build long-lasting and accurate detection systems. However, an optimizer ensemble approach in [21] was introduced, using genetic algorithms (GAs) to fine-tune the parameters of ensemble classifiers for Android malware detection. Evaluated on a large dataset of malware and benign apps, the method achieved superior performance and outperformed other conventional machine learning models. Moreover, [22] proposed an efficient malware

detection framework using traditional machine learning classifiers, including random forest, KNNs, and extra trees, applied to the UNSW-NB15 dataset. By employing feature encoding and an entropy feature selection technique, their approach achieved notable classification performance with random forest.

Despite these advancements, a clear research gap remains. It is observed that a substantial amount of research has been conducted on malware detection using machine learning. However, there are only a few contributions to the literature that explore the use of genetic algorithms in conjunction with current machine learning algorithms for effectively identifying malware in web applications over insecure networks (Table 1).

Table 1. Overview of significant research on learning-based malware detection.

Work	Focus of Research	Method	Description
[16]	Portable executable header features	Feature selection	Excluded non-contributory features to improve model performance
[17]	Detecting unknown malware	Frequency of opcode occurrence	Developed a framework using machine learning techniques
[18]	Cloud-based malware detection	Dataset with GAN-based augmentation	Dynamic and static feature-rich malware classifier using deep learning
[19]	Multimodal malware detection and robustness	CNN-based fusion of grayscale and adversarial defense	Proposed a fusion model leveraging multiple visual malware features
[20]	Enduring malware detection methods	Concept drift	Established that discussing concept drift enhances detection methods
[21]	Android malware classification	Optimizer ensemble learning with GAs	Used GAs to optimize RF parameters for maximum classifier accuracy
[22]	Network-based malware detection	Multiple ML algorithms with TFIDF feature selection	Used KNNs, RF, ET, DT, LR, and nnMLP

3. Proposed Methodology

In this study, we develop a new intelligent malware detection framework specifically designed for secure industrial web applications. Unlike most traditional methods that apply static features or even fall back to black-box models, this approach (MAL-XSEL) combines explainable artificial intelligence and a stacking-based ensemble learning strategy to produce higher accuracy and interpretability. The novelty of this framework consists of a combination of various machine learning models, feature-specific interpretability techniques, such as SHAP and LIME, and a modular workflow that considers the unique structure of portable executables (PEs), as illustrated in Figure 3. A PE file is the standard format for Windows desktop applications. This study targets malware detection by analyzing the PE header, which contains essential metadata and operational information needed by the system to execute the file. By focusing on this structure, the proposed approach aims to detect previously unseen malware that often escapes conventional antivirus systems.

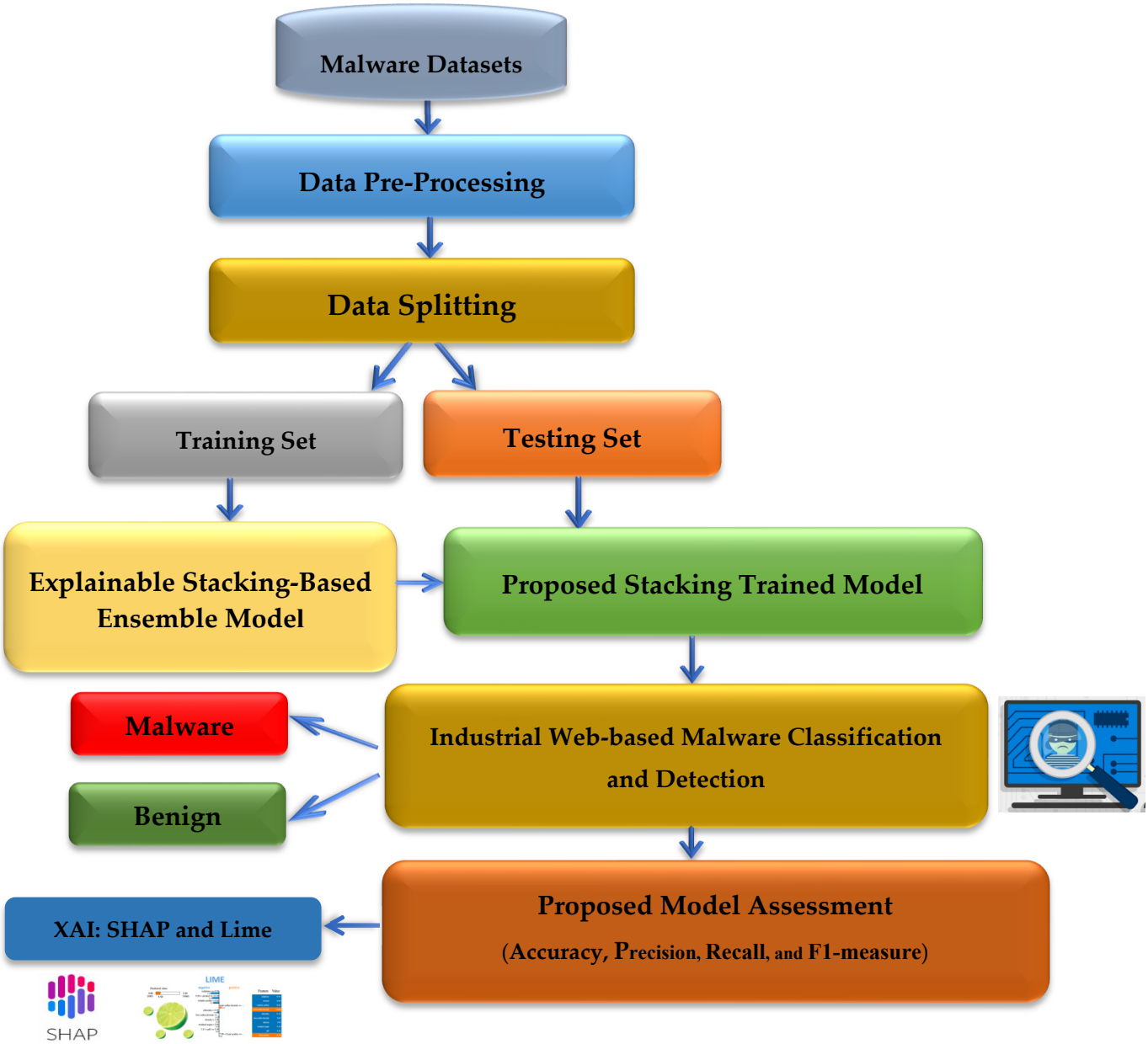


Figure 3. Explainable stacking-based ensemble model for detecting malware.

Furthermore, the proposed MAL-XSEL framework encompasses several key steps essential for the malware recognition process, which are outlined as follows:

1. **Data Collection and Preparation:** Web-based malware datasets are collected from various sources, ensuring a diverse and comprehensive dataset for training and evaluation. The collected data undergo preprocessing, including cleaning, handling missing values, and standardization, to ensure compatibility with the framework's processing requirements.
2. **Data Formatting:** The two selected malware datasets are structured and formatted to align with the requirements of the explainable stacking-based ensemble model. Proper formatting ensures data consistency and enhances feature extraction for improved classification accuracy.
3. **Dataset Partitioning:** The datasets are split into training and testing sets using an 80/20 ratio, where 80% of the data are used to train the model and 20% are reserved for

evaluating its performance. This split ensures an effective balance between training the model and assessing its generalization ability.

4. **Model Training (Explainable Stacking-Based Ensemble Learning):** The core of the proposed approach involves training an ensemble model using a stacking-based strategy. This model integrates multiple machine learning classifiers, such as LightGBM, random forest (RF), CatBoost, decision tree (DT), AdaBoost, k-nearest neighbors (KNN), and linear discriminant analysis (LDA), to leverage their strengths and enhance malware detection accuracy. The training process optimally combines these classifiers to improve predictive performance.
5. **Testing and Classification:** Once trained, the model classifies web-based malware in real time using the 20% test data. The performance of individual base classifiers and the stacking ensemble is analyzed to determine their effectiveness in identifying malicious web-based threats.
6. **Performance Evaluation:** A comprehensive evaluation of the proposed system is conducted using classification-oriented metrics, including accuracy, precision, recall, F1-score, and AUC-ROC. The results are compared against conventional machine learning methods to validate the superiority of the stacking-based ensemble approach.
7. **Explainable AI (XAI) for Model Interpretability:** To enhance transparency and provide insights into the model's decision-making, SHAP is employed to offer both global and local interpretability, explaining each feature's contribution to predictions. Additionally, LIME assesses the impact of individual input features, even identifying minor factors influencing malware detection. This interpretability empowers cybersecurity professionals to better understand and trust the model's classification decisions.

This comprehensive pipeline enables accurate, explainable, and generalizable malware detection, setting MAL-XSEL apart from prior work focused on isolated models or non-interpretable detection strategies.

4. Experimental Study

This section provides an overview of the experimental setup, including details about the two malware datasets utilized. Additionally, it presents the analysis and discussion of the results obtained from the proposed framework, which focuses on classifying malware in executable files within web applications over unsecured systems and assessing the efficacy of the Python (3.11.12)-coded framework used in this study for malware detection.

4.1. Malware Datasets

The projected approach is instigated using numerous PE header datasets containing both malware and benign files sourced from multiple repositories. These datasets are as follows:

1. **Dataset 1 (Classification of Malware (CLaMP)):** This dataset was obtained from the Mendeley data repository. This dataset comprises 5210 file samples, with 2722 classified as malware and 2488 as benign, featuring a total of 69 attributes [23].
2. **Dataset 2 (MalwareDataSet):** This dataset was sourced from GitHub, containing 137,444 samples, of which 40,918 are malware and 96,526 are benign. This dataset includes nine extracted features [24].

4.2. Assessment Criteria

The evaluation of the suggested approach is based on precision, recall, and F-measures, which serve as performance metrics for the classifiers. These metrics are calculated for both positive ("P") and negative ("N") classifications across the test dataset. Table 2 illustrates a confusion matrix (CM), a fundamental tool for evaluating the classification

model's performance. It involves four essential parameters: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). Understanding these parameters aids in comprehensively assessing the classification model's efficacy in detecting malware within web applications in an effective way.

Table 2. CM for malware detection.

	Predicted Benign	**Predicted Malware**
Actual Benign	*TP*	*FN*
Actual Malware	*FP*	*TN*

The following formulas may be used to compute assessment measures such as accuracy, precision, recall, and F1-score once the confusion matrix's parameters have been established [25,26]:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (1)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

$$\text{F1} - \text{score} = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

4.3. Results Analysis

This section discusses the results of the proposed approach for detecting malware. Table 3 presents the performance evaluation of different machine learning models used in the MAL-XSEL system on CLaMP (Dataset 1), based on four key metrics: accuracy, precision, recall, and F1-score. The results indicate that the stacking ensemble model achieves the highest performance, with an accuracy of 0.9962, confirming the effectiveness of combining multiple base learners for improved malware detection. Among the individual models, LightGBM (0.9952 accuracy), CatBoost (0.9942 accuracy), and random forest (RF) (0.9933 accuracy) perform exceptionally well, showcasing the strength of tree-based ensemble techniques in malware classification. Decision tree (DT) (0.9789 accuracy) and AdaBoost (0.9683 accuracy) demonstrate moderate performance but are outperformed by more advanced boosting models such as LightGBM and CatBoost. On the other hand, k-nearest neighbors (KNN) (0.928 accuracy) and linear discriminant analysis (LDA) (0.7639 accuracy) achieve the lowest accuracy, suggesting that traditional models may struggle with complex malware detection patterns. Overall, the results confirm that the stacking-based approach significantly enhances classification accuracy, making MAL-XSEL a highly effective and interpretable solution for web-based malware detection.

Table 4 presents the performance evaluation of the MAL-XSEL system on Malware-DataSet (Dataset 2), utilizing accuracy, precision, recall, and F1-score as key performance metrics. The findings reveal that the stacking ensemble model achieves the highest accuracy (0.9916), reinforcing its effectiveness in improving malware detection. Among individual models, decision tree (DT) (0.9889 accuracy), LightGBM (0.9881 accuracy), and CatBoost (0.9866 accuracy) exhibit strong classification performance, highlighting their robustness in malware detection. Meanwhile, random forest (RF) (0.9758 accuracy) and AdaBoost (0.9694 accuracy) perform moderately well but fall short of the accuracy levels achieved by more advanced boosting techniques. K-nearest neighbors (KNN) (0.9844 accuracy) provides competitive results, though it remains slightly less effective than tree-based mod-

els. Linear discriminant analysis (LDA) (0.8548 accuracy) records the lowest accuracy, suggesting its limitations in handling complex malware classification tasks. Overall, these results confirm that the stacking-based ensemble model delivers the most accurate and balanced performance.

Table 3. Results of the proposed system on Dataset 1.

Model	Accuracy	Precision	Recall	F1-Score
LDA	0.7639	0.7706	0.766	0.7633
KNN	0.928	0.928	0.9279	0.928
DT	0.9789	0.9793	0.9786	0.9789
RF	0.9933	0.9935	0.9931	0.9933
AdaBoost	0.9683	0.9685	0.9682	0.9683
LightGBM	0.9952	0.9953	0.9951	0.9952
CatBoost	0.9942	0.9943	0.9942	0.9942
Stacking	**0.9962**	**0.9962**	**0.9961**	**0.9962**

Table 4. Results of the proposed system on Dataset 2.

Model	Accuracy	Precision	Recall	F1-Score
LDA	0.8547	0.8253	0.8388	0.8314
KNN	0.9843	0.9824	0.9804	0.9814
DT	0.9889	0.9870	0.9867	0.9868
RF	0.9758	0.9698	0.9730	0.9714
AdaBoost	0.9694	0.9686	0.9584	0.9633
LightGBM	0.9881	0.9857	0.9861	0.9859
CatBoost	0.9866	0.9844	0.9838	0.9841
Stacking	**0.9915**	**0.9897**	**0.9902**	**0.9899**

Table 5 presents the class-wise performance of the proposed MAL-XSEL system on Dataset 1, evaluating each model's precision, recall, and F1-score for Class 0 (benign files) and Class 1 (malware files). The stacking ensemble model achieves the highest performance across both classes, with a precision of 0.998 (Class 0) and 0.9944 (Class 1), recall of 0.9941 (Class 0) and 0.9981 (Class 1), and F1-scores of 0.996 (Class 0) and 0.9963 (Class 1). These results highlight the stacking model's superior ability to correctly classify both benign and malware files with minimal misclassification. Among individual classifiers, LightGBM, CatBoost, and random forest (RF) perform exceptionally well, with precision, recall, and F1-scores above 0.99 for both classes, indicating strong generalization and robustness. Decision tree (DT) and AdaBoost follow closely behind, achieving slightly lower but still competitive scores, particularly in recall and F1-score. On the other hand, k-nearest neighbors (KNN) and linear discriminant analysis (LDA) exhibit lower performance. LDA has the weakest performance, particularly in recall for Class 1 (0.6922), indicating that it struggles to correctly detect malware samples. KNN performs better but is still outperformed by tree-based models, reinforcing the effectiveness of ensemble learning for malware classification.

Generally, these results confirm that the stacking-based approach significantly enhances malware detection accuracy for both classes, reducing the risk of false positives (incorrectly classifying benign files as malware) and false negatives (failing to detect actual malware). This further solidifies MAL-XSEL as a highly reliable and interpretable solution for web malware detection.

Table 5. Results of the proposed system on Dataset 1 for Class 0 and Class 1.

Model	Precision		Recall		F1-Score	
	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1
LDA	0.7203	0.8208	0.8399	0.6922	0.7755	0.751
KNN	0.9267	0.9292	0.9249	0.931	0.9258	0.9301
DT	0.9879	0.9707	0.9684	0.9888	0.978	0.9797
RF	0.998	0.9889	0.9881	0.9981	0.993	0.9935
AdaBoost	0.9721	0.9649	0.9625	0.9739	0.9672	0.9694
LightGBM	0.998	0.9926	0.9921	0.9981	0.995	0.9953
CatBoost	0.996	0.9926	0.9921	0.9963	0.9941	0.9944
Stacking	**0.998**	**0.9944**	**0.9941**	**0.9981**	**0.996**	**0.9963**

Table 6 provides a class-wise evaluation of the MAL-XSEL system on Dataset 2, analyzing the precision, recall, and F1-score of each model for Class 0 (benign files) and Class 1 (malware files). The stacking ensemble model demonstrates the highest overall performance, achieving a precision of 0.9943 (Class 0) and 0.9852 (Class 1), recall of 0.99358 (Class 0) and 0.9868 (Class 1), and F1-scores of 0.9939 (Class 0) and 0.9860 (Class 1). These results highlight the stacking model's superior capability in accurately distinguishing between benign and malware samples, reinforcing its reliability as a robust classification approach. Among individual classifiers, LightGBM, CatBoost, and random forest (RF) deliver outstanding results, with precision, recall, and F1-scores exceeding 0.98 for both classes, demonstrating strong generalization across different malware variations. Decision tree (DT) and AdaBoost also perform well, though AdaBoost exhibits a slightly lower recall for Class 1 (0.9306), suggesting a tendency to misclassify some malware samples. In contrast, k-nearest neighbors (KNN) and linear discriminant analysis (LDA) show comparatively weaker performance. LDA, in particular, records the lowest scores, particularly for Class 1 (precision: 0.7407, recall: 0.79874, F1-score: 0.7686), indicating its challenges in accurately classifying malware. While KNN outperforms LDA, it remains less effective than ensemble-based models, reaffirming the advantage of stacking and boosting techniques for malware detection. Overall, these findings confirm that the stacking-based ensemble model significantly enhances malware detection for both benign and malicious files, reducing misclassification risks. The results further establish MAL-XSEL as a powerful and interpretable cybersecurity solution, offering enhanced security for web applications against evolving malware threats.

Table 6. Results of the proposed system on Dataset 2 for Class 0 and Class 1.

Model	Precision		Recall		F1-Score	
	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1
LDA	0.9098	0.7407	0.87902	0.79874	0.8941	0.7686
KNN	0.9852	0.9545	0.98008	0.96603	0.9826	0.9602
DT	0.9872	0.9775	0.99035	0.97049	0.9888	0.9740
RF	0.9918	0.9821	0.99228	0.9812	0.9920	0.9816
AdaBoost	0.9704	0.9668	0.98618	0.9306	0.9782	0.9483
LightGBM	0.9917	0.9797	0.99124	0.9809	0.9915	0.9803
CatBoost	0.9899	0.9788	0.99087	0.9767	0.9904	0.9778
Stacking	**0.9943**	**0.9852**	**0.99358**	**0.9868**	**0.9939**	**0.9860**

Likewise, Figures 4 and 5 show the confusion matrices for Dataset 1 and Dataset 2, respectively. Figure 6 presents a comparison of the results for Dataset 1 and Dataset 2 using the proposed approach and other machine learning models. Stacking outperforms all other models on both datasets, achieving the highest accuracy (0.9915 on Dataset 1 and 0.9962 on Dataset 2), along with superior precision, recall, and F1-scores. This confirms that the stacking model is the most effective approach compared to LDA, KNN, DT, RF, AdaBoost, LightGBM, and CatBoost.

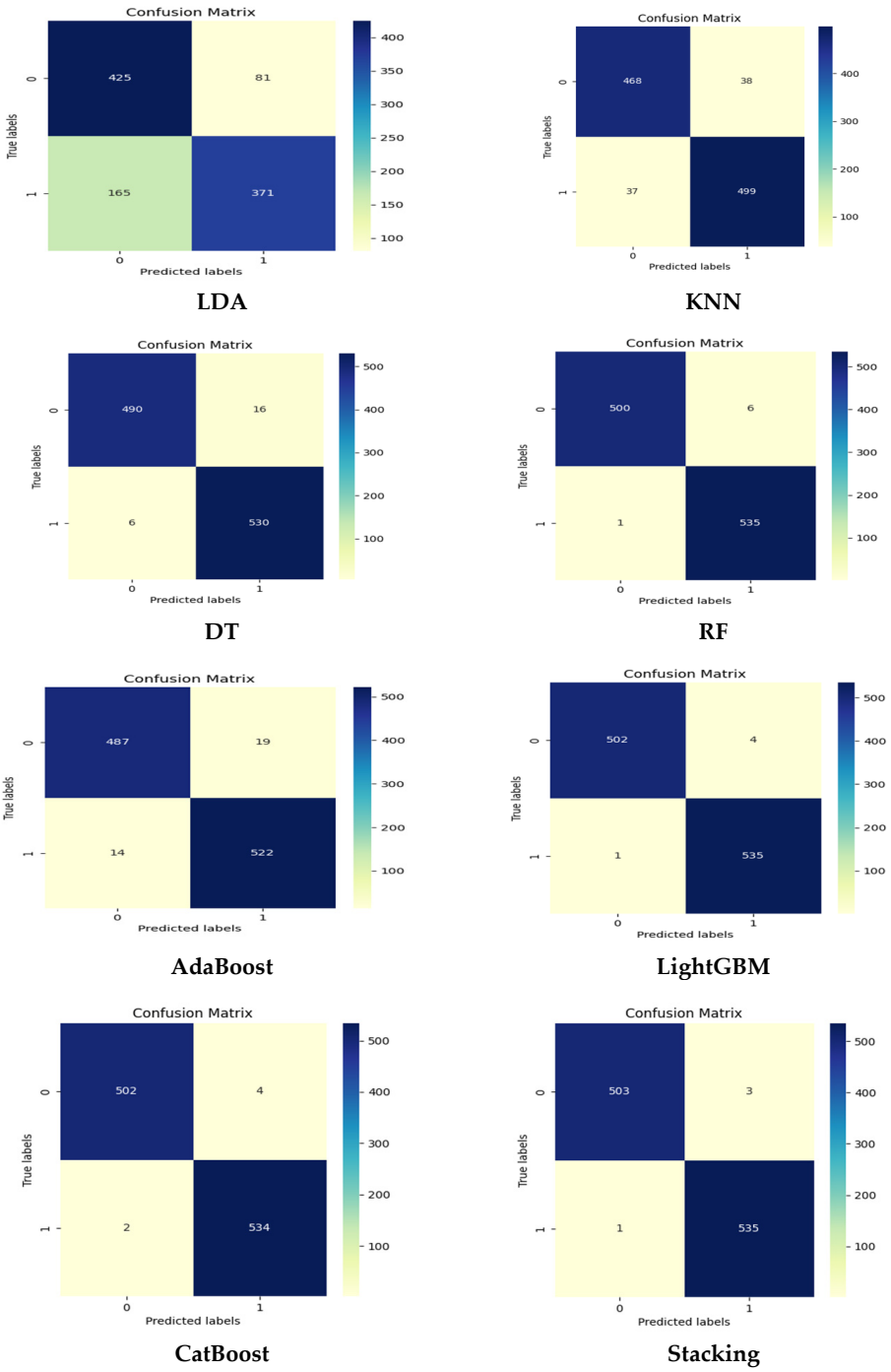


Figure 4. Confusion matrix for Dataset 1.

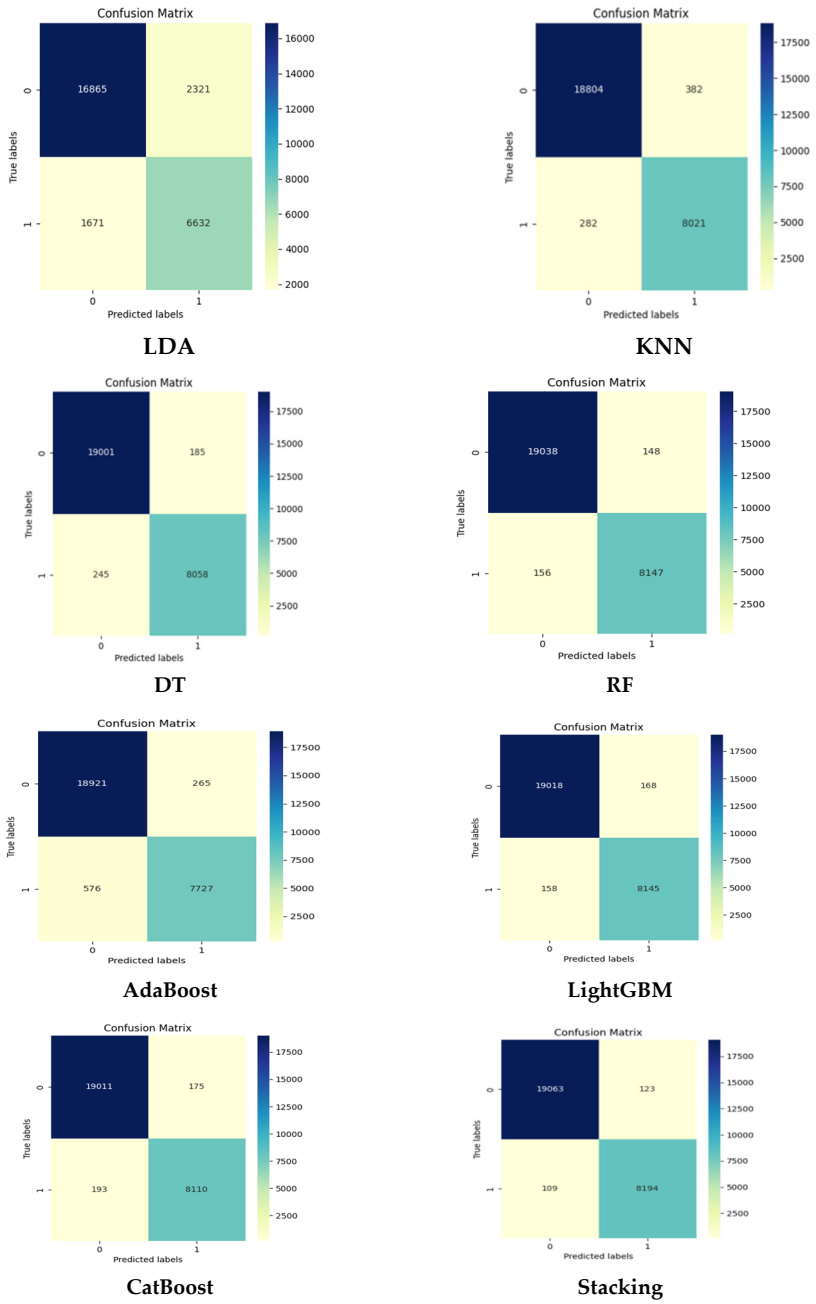


Figure 5. Confusion matrix for Dataset 2.

Interpretability enhances confidence in AI models by clarifying predictions and exposing model weaknesses. In this work, it is wanted to achieve a transparent analysis of the classification decisions by assessing individual features by means of XAI techniques—SHAP and LIME—as shown in Figures 7–9. These visual explanations help cybersecurity analysts better understand why a sample was classified as malware versus benign, allowing for better decision-making and thereby facilitating policy adjustments. For example, Figure 8 features SHAP summary plots that highlight specific features, for instance, Feature 3 or Feature 5, that greatly affect the classification decision in both datasets. In the same way, Figure 9 ranks feature importance based on average impact so that it guides the analysts on which behaviors or signals to watch most closely. Our framework enhances real-time applications by providing such insights (e.g., adaptive firewall rules, automatic alert thresholds, or even policy writing for industrial intrusion prevention systems (IPSs)). In using such interpretability tools, we are effectively converting the raw predictions into actionable

intelligence, which will enhance the usability of the system, making it more practical for real-world cybersecurity implementations.

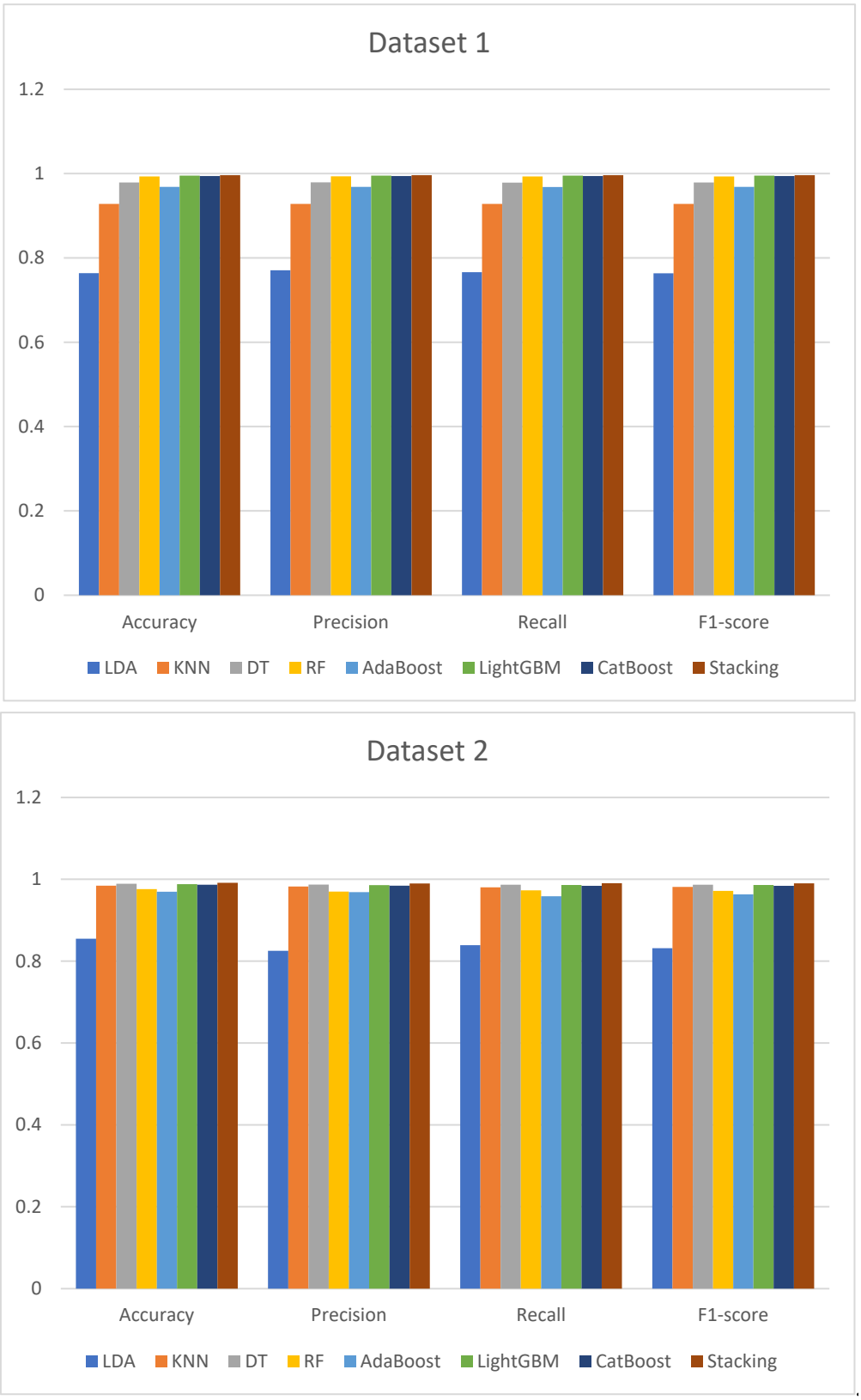
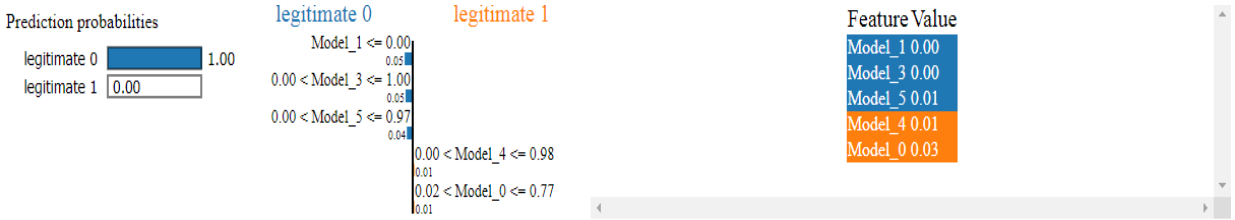


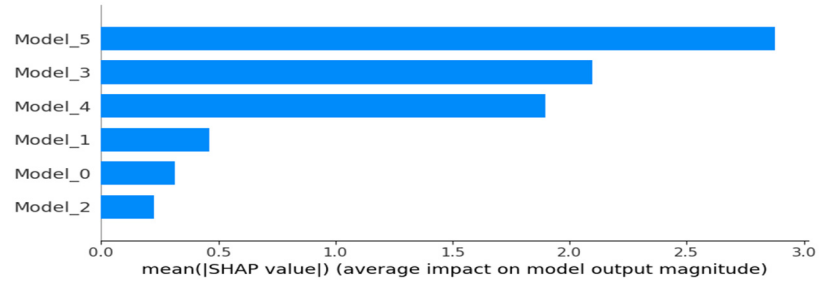
Figure 6. A comparison of the proposed approach and other ML models for Dataset 1 and Dataset 2 for malware detection.

(a) Dataset 1

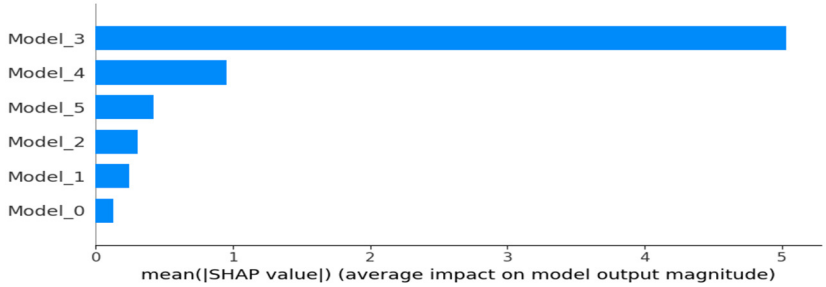


(b) Dataset 2

Figure 7. Lime for Dataset 1 and Dataset 2 with the proposed stacking approach.



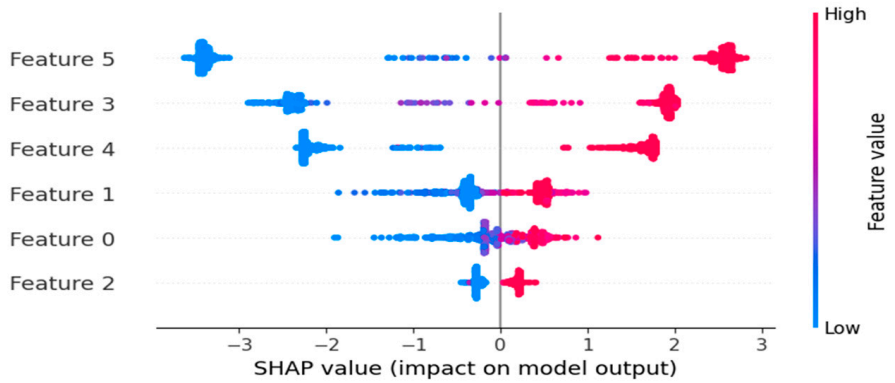
(a) Dataset 1



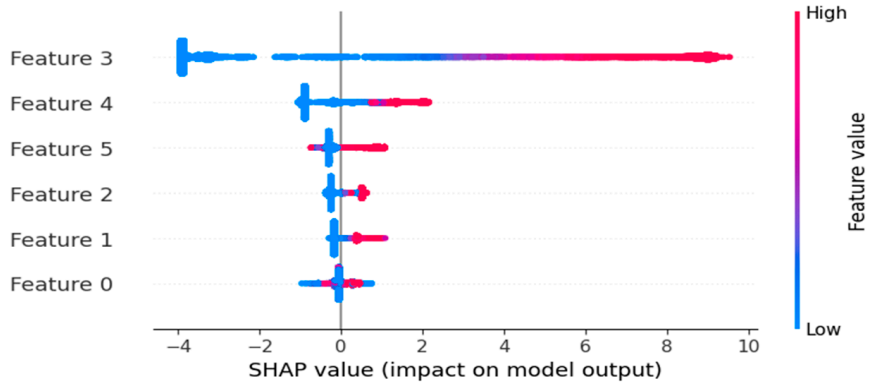
(b) Dataset 2

Figure 8. Shape for model impact for Dataset 1 and Dataset 2 with the proposed stacking approach.

Table 7 and Figure 10 present a comparative analysis between the proposed MAL-XSEL framework and an existing malware detection model [27] based on their performance on two datasets: ClaMP (Dataset 1) and MalwareDataSet (Dataset 2). The comparison evaluates accuracy as the primary performance metric. The existing approach utilizes a 1D-CNN-LSTM model, a deep learning-based hybrid architecture, achieving an accuracy of 98.75% on the ClaMP dataset and 97.07% on the MalwareDataSet dataset. While these results indicate strong performance, deep learning models often operate as "black boxes", lacking interpretability, which can limit their adoption in security-critical applications.



(a) Dataset 1



(b) Dataset 2

Figure 9. Shape for feature importance for Dataset 1 and Dataset 2 with the proposed stacking approach.

Table 7. Comparison between the proposed framework and existing work.

Work	Dataset	Model	Outcomes
[27]	ClaMP	1D-CNN-LSTM Model	Accuracy is 98.75%
	MalwareDataSet		Accuracy is 97.07%
Proposed Approach	ClaMP (Dataset 1)	Stacking Ensemble Approach	Accuracy is 99.62%
	MalwareDataSet (Dataset 2)		Accuracy is 99.15%

In contrast, the proposed approach employs a hybrid stacking-based ensemble of machine learning models, significantly outperforming the existing work. It achieves an accuracy of 99.62% on the ClaMP dataset and 99.15% on the MalwareDataSet dataset, demonstrating a notable improvement in malware classification. The stacking ensemble approach leverages multiple base learners, optimizing classification performance by combining the strengths of different machine learning models. In general, this comparison highlights the effectiveness of the proposed MAL-XSEL framework, which not only improves accuracy but also incorporates explainability through XAI techniques. By enhancing transparency in decision-making, MAL-XSEL provides a more robust and interpretable solution for malware detection in industrial web applications, ensuring greater security for critical infrastructure, industrial control systems, and enterprise networks.

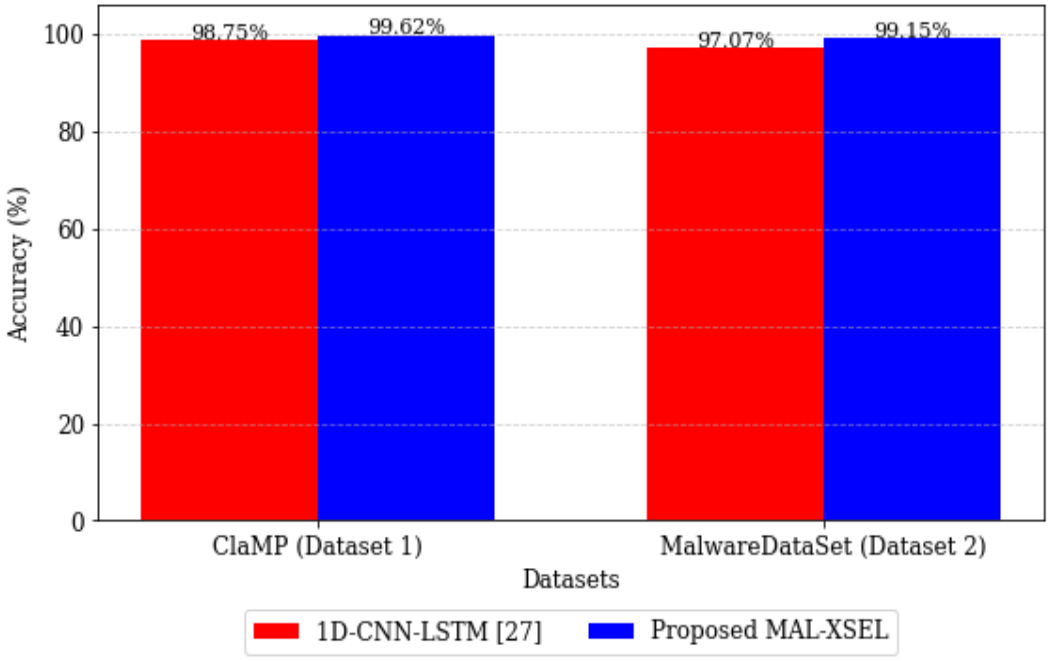


Figure 10. Accuracy comparison of the proposed technique with related work.

4.4. Limitations and Challenges

Despite the promising results achieved by the proposed **MAL-XSEL** framework, several limitations and practical challenges remain that warrant further investigation for real-world deployment, particularly in industrial control systems (ICSs) and IIoT environments. These limitations can be summarized as follows:

- **Computational Complexity:** The stacking ensemble model, while achieving high accuracy, requires significant computational resources during training and inference. In industrial environments, where real-time threat detection is crucial for operational continuity, the integration of multiple base learners and an additional meta-learner increases processing time, making immediate malware detection in resource-constrained industrial control systems (ICSs) and IIoT networks challenging.
- **Feature Dependence and Data Variability:** The model's performance is influenced by the quality and diversity of the datasets used for training. In industrial sectors, malware targeting ICSs and enterprise networks often exhibit distinct characteristics compared to traditional IT environments. Differences in feature representation across various malware datasets may affect generalizability, requiring extensive feature engineering and adaptation for new or unseen malware types.
- **Interpretability vs. Performance Trade-off:** While SHAP and LIME enhance interpretability, their computational cost can be high, especially when explaining complex ensemble models. Generating local and global explanations for large-scale datasets can slow down decision-making in industrial security operation centers. This trade-off must be carefully balanced to maintain operational efficiency.
- **Dataset Generalization:** The datasets used in this study vary in size and composition, which may limit the generalizability of the model across all real-world malware threats. Additionally, zero-day malware or novel attack vectors may not be well-represented, which poses challenges for detection. Addressing this requires ongoing model adaptation and the inclusion of more diverse and dynamic data sources.
- **Adversarial Evasion and Concept Drift:** Malware variants continuously evolve, and adversarial attacks may attempt to bypass detection models by obfuscating features. Industrial environments, such as power grids, smart factories, and IIoT systems,

are particularly vulnerable to these evolving threats. Additionally, concept drift in malware characteristics over time necessitates frequent retraining and adaptation to maintain model effectiveness.

- **Scalability and Deployment Challenges:** Deploying the MAL-XSEL system in large-scale, real-world ICS or IIoT environments introduces several practical challenges that must be addressed to ensure operational feasibility. To maintain adaptability in dynamic threat landscapes, the system should support incremental learning or modular retraining to effectively handle newly emerging malware variants. Additionally, compliance with cybersecurity standards and interoperability with various industrial components are essential to ensure the system's credibility, security, and scalability. Low-latency, real-time inference is also critical for effective threat detection in industrial systems, where delays can result in significant operational risks. To address the computational burden, optimization techniques such as model pruning, quantization, and knowledge distillation can be applied to reduce the model's footprint. In this context, edge computing presents a promising solution by enabling local decision-making with real-time responsiveness, thereby reducing network latency and ensuring timely malware detection.

5. Conclusions and Future Work

In this study, we proposed MAL-XSEL, an explainable stacking-based ensemble learning model for web malware classification. By integrating explainable artificial intelligence (XAI) techniques, the proposed framework enhances both detection accuracy and interpretability, addressing the limitations of traditional "black-box" malware detection models. In industrial environments, where cybersecurity is critical for protecting industrial control systems (ICSs), IIoT networks, and enterprise web applications, ensuring transparency in malware detection is essential. Through the application of SHAP and LIME, security professionals can gain deeper insights into the decision-making process, improving trust and adoption in security-critical environments. The experimental evaluation on two benchmark datasets demonstrated the superior performance of MAL-XSEL compared to conventional machine learning models. The stacking ensemble approach consistently achieved higher accuracy, with a peak accuracy of 0.9962 in Dataset 1 and 0.9916 in Dataset 2, confirming its effectiveness in malware detection. These results validate the potential of ensemble learning and XAI techniques in strengthening cybersecurity defenses for web applications. By combining high detection accuracy, transparency, and interpretability, MAL-XSEL represents a practical and robust solution for fighting the evolving malware threats in industrial systems.

Future research can focus on reducing complexity for real-time deployment in resource-constrained environments or improving the adaptability of models to malware behavior changes due to concept drift. However, developing lightweight models that are interpretable yet maintain a high detection accuracy will also be critical for industrial-scale implementation. In the future, to increase the efficiency of systems against zero-day threats, one can integrate live traffic streaming datasets or live data streams with real network simulations. Applying model simplification approaches can be studied with the intention of minimizing the resource requirements of the model while ensuring performance and transparency. In the same context, implementing post hoc XAI techniques during the critical analysis rather than during real-time applications would also strike a balance between interpretability and latency. These paths are going to help reduce the gap between detection accuracy and operational deployment, thereby enhancing the prospect of explainable AI for applications within the realms of cybersecurity in practice.

Author Contributions: Conceptualization, E.E.-D.H., S.A., O.A.G. and A.S.; Formal analysis, E.E.-D.H. and A.S.; Funding acquisition, S.A.; Investigation, H.E., O.A.G. and A.S.; Methodology, H.E.; Software, E.E.-D.H. and O.A.G.; Validation, H.E.; Writing—original draft, E.E.-D.H. and A.S.; Writing—review and editing, S.A., H.E. and O.A.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2025R197), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Data Availability Statement: The original contributions presented in this study are included in the article; further inquiries can be directed to the corresponding author.

Acknowledgments: Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2025R197), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviation

AI	Artificial Intelligence
AUC-ROC	Area Under the Curve—Receiver Operating Characteristic
PE	Portable Executables
CM	Confusion Matrix
CNN	Convolutional Neural Network
DT	Decision Tree
FN	False Negative
FP	False Positive
ICS	Industrial Control System
IIoT	Industrial Internet of Things
KNN	K-Nearest Neighbors
LDA	Linear Discriminant Analysis
LIME	Local Interpretable Model-Agnostic Explanations
RF	Random Forest
SHAP	SHapley Additive exPlanations
XAI	Explainable Artificial Intelligence

References

1. Egele, M.; Scholte, T.; Kirda, E.; Kruegel, C. A survey on automated dynamic malware-analysis techniques and tools. *ACM Comput. Surv. (CSUR)* **2008**, *44*, 1–42. [CrossRef]
2. Falowo, I.; Ozer, M.; Li, C.; Abdo, J.B. Evolving Malware and DDoS Attacks: Decadal Longitudinal Study. *IEEE Access* **2024**, *12*, 39221–39237. [CrossRef]
3. Knapp, E.D. *Industrial Network Security: Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems*; Elsevier: Amsterdam, The Netherlands, 2024.
4. Al-Hawawreh, M.; Alazab, M.; Ferrag, M.A.; Hossain, M.S. Securing the Industrial Internet of Things against ransomware attacks: A comprehensive analysis of the emerging threat landscape and detection mechanisms. *J. Netw. Comput. Appl.* **2024**, *223*, 103809. [CrossRef]
5. Zhang, X.; Wu, K.; Chen, Z.; Zhang, C. MalCaps: A Capsule Network Based Model for the Malware Classification. *Processes* **2021**, *9*, 929. [CrossRef]
6. Zhao, S.; Tuan, L.A.; Fu, J.; Wen, J.; Luo, W. Exploring Clean Label Backdoor Attacks and Defense in Language Models. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2024**, *32*, 3014–3024. [CrossRef]
7. Takemoto, K. All in How You Ask for It: Simple Black-Box Method for Jailbreak Attacks. *Appl. Sci.* **2024**, *14*, 3558. [CrossRef]
8. Cui, X.; Aparcedo, A.; Jang, Y.K.; Lim, S.N. On the Robustness of Large Multimodal Models against Image Adversarial Attacks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–18 June 2024; pp. 24625–24634.

9. Jiang, W.; Han, H.; He, M.; Gu, W. ML-based pre-deployment SDN performance prediction with neural network boosting regression. *Expert. Syst. Appl.* **2024**, *241*, 122774. [CrossRef]
10. Jiang, W. Graph-based deep learning for communication networks: A survey. *Comput. Commun.* **2022**, *185*, 40–54. [CrossRef]
11. Lu, H.; Uddin, S. Explainable stacking-based model for predicting hospital readmission for diabetic patients. *Information* **2022**, *13*, 436. [CrossRef]
12. Kovanen, T.; Nuojua, V.; Lehto, M. Cyber threat landscape in energy sector. In Proceedings of the ICCWS 2018 13th International Conference on Cyber Warfare and Security, Washington, DC, USA, 8–9 March 2018; Academic Conferences and Publishing Limited: Cambridge, MA, USA, 2018; p. 353.
13. Huang, Y.; Liu, J.; Xiang, X.; Wen, P.; Wen, S.; Chen, Y.; Chen, L.; Zhang, Y. Malware Identification Method in Industrial Control Systems Based on Opcode2vec and CVAE-GAN. *Sensors* **2024**, *24*, 5518. [CrossRef]
14. Zheng, Y.; Na, Z.; Ji, W.; Lu, Y. An Adaptive Fuzzy SIR Model for Real-Time Malware Spread Prediction in Industrial Internet of Things Networks. *IEEE Internet Things J.* **2025**. [CrossRef]
15. Naeem, H.; Ullah, F.; Naeem, M.R.; Khalid, S.; Vasan, D.; Jabbar, S.; Saeed, S. Malware detection in industrial internet of things based on hybrid image visualization and deep learning model. *Ad. Hoc Netw.* **2020**, *105*, 102154. [CrossRef]
16. Hussain, A.; Asif, M.; Ahmad, M.B.; Mahmood, T.; Raza, M.A. Malware detection using machine learning algorithms for windows platform. In Proceedings of the International Conference on Information Technology and Applications: ICITA 2021, Dubai, United Arab Emirates, 13–14 November 2021; Springer Nature: Singapore, 2022; pp. 619–632.
17. Sharma, S.; Rama Krishna, C.; Sahay, S.K. Detection of advanced malware by machine learning techniques. In *Soft Computing: Theories and Applications: Proceedings of the SoCTA 2017*; Springer: Singapore, 2019; pp. 333–342.
18. Nguyen, P.S.; Huy, T.N.; Tuan, T.A.; Trung, P.D.; Long, H.V. Hybrid Feature Extraction and Integrated Deep Learning for Cloud-Based Malware Detection. *Comput. Secur.* **2025**, *150*, 104233. [CrossRef]
19. Johny, J.A.; Asmitha, K.A.; Vinod, P.; Radhamani, G.; Rehiman, K.A.R.; Conti, M. Deep learning fusion for effective malware detection: Leveraging visual features. *Clust. Comput.* **2025**, *28*, 135. [CrossRef]
20. Guerra-Manzanares, A.; Bahsi, H.; Luckner, M. Leveraging the first line of defense: A study on the evolution and usage of android security permissions for enhanced android malware detection. *J. Comput. Virol. Hacking Tech.* **2023**, *19*, 65–96. [CrossRef]
21. Taha, A.; Barukab, O. Android malware classification using optimized ensemble learning based on genetic algorithms. *Sustainability* **2022**, *14*, 14406. [CrossRef]
22. Azeem, M.; Khan, D.; Iftikhar, S.; Bawazeer, S.; Alzahrani, M. Analyzing and Comparing the Effectiveness of Malware Detection: A Study of Machine Learning Approaches. *Heliyon* **2024**, *10*, e24008. [CrossRef] [PubMed]
23. Kumar, A. ClaMP (Classification of Malware with PE Headers). Mendeley Data 2020, V1. Available online: https://data.mendeley.com/datasets/xvyv59vwvz/1 (accessed on 18 January 2025).
24. Yıldırım, E. MalwareDataSet. Available online: https://github.com/emr4h/Malware-Detection-Using-Machine-Learning (accessed on 18 January 2025).
25. Alshathri, S.; Sayed, A.; Hemdan, E.E.-D. An Intelligent Attack Detection Framework for the Internet of Autonomous Vehicles with Imbalanced Car Hacking Data. World Electr. *Veh. J.* **2024**, *15*, 356. [CrossRef]
26. Sayed, A.; Alshathri, S.; Hemdan, E.E.-D. Conditional Generative Adversarial Networks with Optimized Machine Learning for Fault Detection of Triplex Pump in Industrial Digital Twin. *Processes* **2024**, *12*, 2357. [CrossRef]
27. Alqahtani, A.; Azzony, S.; Alsharafi, L.; Alaseri, M. Web-Based Malware Detection System Using Convolutional Neural Network. *Digital* **2023**, *3*, 273–285. [CrossRef]