Integrating Explainable AI (XAI) for Improved Malware Detection and Analysis

MSc Research Project
MSc in Cybersecurity

Sneha Sivaram
Student ID: X23192054

School of Computing
National College of Ireland

Supervisor: Prof. Liam Mccabe

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……. ………………………..Sneha Sivaram…………………………………………………

Student ID: …………………………………X23192054………………………………………………..……

Programme: ……………Msc in Cybersecurity………………… **Year:** ……………2024…………….**.**

Module: ………………………………………………MSc Research Project……………………………

Supervisor: ………………………………………Prof. Liam Mccabe……………………………………………

Submission Due Date: ……………………………….12/12/2024…………………………………………………….………

Project Title: Integrating Explainable AI (XAI) for Improved Malware Detection and Analysis

Word Count: ………………8688………………… **Page Count**……………………29………**.**……**..**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ………………………..Sneha Sivaram………………………………………………

Date: ………………………………12/12/2024…………………………………………

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Table of Contents

LIST OF TABLES

LIST OF FIGURES

Integrating Explainable AI (XAI) for Improved Malware Detection and Analysis

Sneha Sivaram

X23192054

Abstract

Malware has significantly evolved over the decades, transitioning from simple viruses to complex threats such as Advanced Persistent Threats (APTs). This evolution requires robust and advanced detection methods. Traditional methods, including signature-based malware detection, struggle with obfuscated and novel malware. This research integrates machine learning (ML) models: Logistic Regression, Support Vector Machine (SVM), and Random Forest with Explainable (XAI) techniques, specifically LIME (Local Interpretable Model-Agnostic Explanations), to improve malware detection system's accuracy and interpretability. Using a malware memory dump dataset, the Logistic Regression model achieved the highest accuracy of 99.94%, while the Random Forest model showed signs of overfitting. To utilise the full potential of this XAI-based malware detection system, an email alert system was incorporated to send alerts to the administrator with proper explanations made by the XAI technique whenever the system detects potential malware.

1 Introduction

Malware attacks have been a persistent threat to cybersecurity since the emergence of computer viruses in the 1980s. Due to the malware's evolving nature, traditional detection mechanisms have become insufficient (ENISA). It is crucial to have an effective and adaptive malware detection mechanism. For example, signature-based malware detection, such as antivirus software, relies on predefined malware patterns and fails against zero-day attacks or novel malware types (Capuano et al., 2022). Machine learning-based malware detection was introduced as it performed better than the signature-based model in detecting malware, as the models could be pre-trained with selected features. However, these models often function as 'black boxes', providing predictions without explaining the reason. This puts security professionals in a difficult position where, although they can utilise advanced detection methods, they still lack interpretability and transparency (Moore et al., 2018).

This is where a new technique called Explainable Artificial Intelligence, also known as XAI enters, where it can address the gap by providing valuable insights into the decisions made by complex machine learning models. In malware detection, XAI offers positive capabilities which not only enhance the accuracy of the predictions but also build trust among security professionals with decision-making (Moore et al., 2018). This study aims to focus on integrating explainable AI techniques into conventional malware detection systems to provide an effective solution for understanding and solving malicious activity in an informed manner. The use of the XAI technique not only guarantees transparency to automated decisions made by the model but also enables security professionals to trust and validate the outputs produced by these systems.

1.1 Research Background

Malware has evolved a lot in the past few decades. From simple viruses in the 1980s to today's sophisticated threats like ransomware, trojan, and Advanced Persistent Threats (APTs), malware has become very sophisticated and damaging (ENISA). Traditional detection systems failed to keep up with evolving malware variants (Capuano et al., 2022). As a result, those traditional methods failed when it came to zero-day attacks which has the ability to attack without any warnings.

To address these concerns, machine learning was adopted for malware detection. These models are capable of recognising various patterns and characteristics that can differentiate malicious software from normal, benign programs (Capuano et al., 2022). Some of those methods were clustering, anomaly detection and supervised learning such as support vector machine and random forest. These have shown positive behaviour, which has led to improving detection rates and accuracy. Even though these models performed successfully, they had a fundamental issue, which is that they performed as black-box models, which means they were not able to provide any explanation as to how they reached their conclusion.

This lack of transparency can definitely affect their deployment in security centres as security professionals are reluctant to act on the alerts made by the models but they are not able to fully understand (Moore et al., 2018) and there are issues of false positives and false negatives which can lead to any false alarms, that might lead to any severe consequences like ignoring an actual threat or flagging a legitimate file as malware So, the user must be able to understand how these systems come to these decisions. Therefore explainable AI is an essential tool to understand these black box models deeply and also make them interpretable and trustable.

1.1.1 Explainable AI and its Role in Malware Detection

Explainable AI refers to a set of methods that helps understand the internal workings of a machine learning model, which is understandable to humans. These methods provide explanations that illustrate the important features or the data points that led to a particular classification, unlike normal machine learning models, which are not able to do the same. These methods not only help in validating whether a model is functioning properly but also provide useful insights that allow security professionals to make the right decisions.

There are two primary categories present under XAI: intrinsic and post-hoc methods. Intrinsic methods are built directly into the model which results in simpler models such as decision trees or linear models, so they are transparent by design. But post-hoc methods Or applied after the model has been properly trained and the techniques are: LIME (Local Interpretable Model-agnostic Explanations) and SHAP (SHapley Additive exPlanations).

(Ribeiro et al., 2016) These methods are used with complex models to create local or global explanations.

In the context of malware detection, XAI provides security professionals with insights into why a specific file was flagged as malicious by highlighting the particular features or behaviours that the model considers suspicious (Moore et al., 2018). This not only adds an extra layer of validation to the model's decision-making process but also helps analysts identify possible false alarms, which can help in making the detection mechanisms more robust.

1.2 Problem Definition

1.2.1 The Challenge of Balancing Accuracy and Interpretability

The primary challenge lies in balancing high detection accuracy with interpretability. Complex models such as neural networks yield very high detection rates but they lack transparency, making them unsuitable for critical applications where trust is important (Capuano et al., 2022). Security professionals require models that:

- Provide accurate predictions.
- Explain the decision-making process clearly.
- Reduce false positives and negatives to prevent erroneous actions.

So, when XAI is used with these machine learning-based malware detection systems, then it can predict clear and human-readable outputs, which will help the security professionals make informed and swift decisions in case a malicious file or an unknown file with suspicious features is detected.

1.2.2 Research Question: How can Explainable AI improve the accuracy and transparency in detecting malware?

Technologies have been rapidly developing through the years, which has led to an increase in malware attacks, which is a serious risk to networks and systems. Since everything is now online and connected, it can be a risk in some fields where it is important to safeguard the data. One of the most common types of cyber threat is a malware attack, which compromises the CIA (Confidentiality, Integrity and Availability) triad of information systems. Malware has many variants and it keeps evolving over the years, so it is necessary to have advanced detection methods as well.

The main goal of this research is to use ML models to improve malware detection capabilities, especially focusing on explainable AI to overcome the interpretability issues and hence, improve the transparency of these models. Traditional detection methods such as signature-based malware detection and antivirus software have been known to struggle to keep up with evolving kinds of hardware as malware evolved through the years, it is important that the detection methods should also be improved. So, in this context machine learning along with explainable AI proves to be a key component in developing malware detection techniques.

Approach and novelty

The research aims to use ML models such as Logistic Regression, Support Vector Machines (SVM) and Random Forest Classifiers integrated with Local Interpretable Model Agnostic Explanations (LIME) to enhance the model's performance by improving its interpretability and complex decision-making process. The use of LIME provides human-understandable explanations for the predictions made by the machine learning models which can be crucial for security professionals to make important decisions in important fields.

The novelty of this research is present in the combined use of ML models with the LIME for post-hoc interpretability to identify and prevent malware. Additionally, the research incorporates a real-time monitoring system with email alerts, which will provide timely notification to security professionals when suspicious activity is detected.

The major contributions are:

- Developing an integrated model that combines LIME with Machine Learning models to ensure accuracy is maintained along with proper interpretability.
- Improving prior research by focusing on one single technique: LIME, which will ensure lightweight and local explanations are given for model predictions without any complex issues.
- Implementing a real-time detection and alert mechanism which will use the model predictions to generate email alerts whenever malware is detected. This feature can be very useful for fast response times.
- Evaluating the effectiveness of XAI technique LIME to fill the gap between model accuracy and transparency.

The existing literature highlights the limitations of black box models, especially the difficulty for security professionals to trust the model outputs without any proper explanations. So, Explainable AI offers a potential solution by opening the black box and providing valuable insights into how each features individually contribute to malware classification predictions. In addition, the use of a real-time alert system based on the LIME integrated model will make sure that security professionals are properly informed about potential malware threats, which will help them react effectively to mitigate risks.

Security considerations

According to the European Union Agency for Cyber Security (ENISA), the complexity of malware attacks has been increasing, so it is important to adopt more advanced malware detection methods which can address the problems faced by current systems. This research aims to address these recommendations by focusing on applying LIME as an interpretable tool to machine learning models for malware detection systems.

The use of explainable AI techniques for malware detection has been further supported by previous studies (Moore et al., 2018) which explain the necessity for integrating interpretable models in cyber security to improve the human understandability of predictions made by machine learning models. It is important to have a trusted system where the security professional can understand how the system came to that decision since it's a matter of security.

In addition, there were significant contributions to this domain from the survey on explainable artificial intelligence in cyber security (Capuano et al., 2022). This work explains different XAI techniques and their use cases in cyber security, which highlighter LIME as one of the most promising methods for providing human-understandable explanations.

Furthermore, some insights indicate that while traditional black box models are known to show high accuracy in malware detection, They lack the transparency needed for real-world deployment in the field of cybersecurity (Ribeiro et al., 2016). These references show the importance of integrating transparent XAI techniques to ensure that the model predictions are trustworthy and validated by security professionals.

Finally, the remainder of this research paper is structured as follows: Section 2 will discuss the previous studies and research in more detail. Section 3 elaborates on the methodology, classification, algorithms, dataset, and data pre-processing. Section 4 will discuss design specifications. Section 5 will explain the implementation. Section 6 presents the evaluation of the results. Section 7 presents the conclusion and future work, and finally, Section 8 includes references.

2 Related Work

In this section, the previous studies and research have been critically reviewed, which highlights significant contributions and the progress they have made, the methodology used and the challenges faced. This review will focus on the importance of explainability, providing a detailed overview of the XAI techniques used in malware detection and how they address the limitations of traditional ML models. In addition, traditional ML-based malware detection work has also been reviewed to show how it's not very effective without explainable AI.

2.1 ML and DL Models for Malware Detection

The use of machine learning and deep learning models for malware detection is very popular due to their potential for identifying complex patterns in data. There is a wide range of studies that explore different approaches and techniques to use ML and DL models effectively for detecting malware that offers results in terms of accuracy, efficiency, and applicability.

The research by (Akhtar and Feng, 2022) '*Malware Analysis and Detection Using Machine Learning Algorithms*' provides an overview of different machine learning models applied in malware detection, such as Random Forest (RF), support vector machine (SVM) and K nearest neighbours (KNN). The authors argue that the traditional signature-based methods fail to adapt to the evolving nature of malware, whereas machine learning models have more adaptability. This study highlighted that random forest provides a strong classification accuracy but lacks interpretability. SVM was also shown to be effective in linear classification problems, but it is known to be computationally expensive when data sets become larger, so this may restrict real-time applicability. Overall, this research paper shows that newer machine-learning techniques must be incorporated to enhance accuracy and interpretability which directly aligns with the goals of this project. This research project aims to address the problem by incorporating XAI techniques to enhance interpretability while maintaining high accuracy.

The research by the author (Alomari et al., 2023) titled '*Malware Detection Using Deep Learning and Correlation-Based Feature Selection*' studies the efficiency of deep learning models such as Convolutional Neural Networks CNNs and Long Short-Term Memory LSTM networks for classifying malware samples. The authors presented their research showing that CNN is very effective in feature extraction from larger datasets, but they require a very high amount of computational power for training, meaning that they are quite costly when it comes to real-world applications. On the other hand, LSTM was found to be better suited for sequence data, especially for detecting malware in network traffic. Although deep learning techniques are shown to improve detection and accuracy significantly, the authors mention that their black box nature limits interpretability, which makes it challenging for security professionals to fully trust the model predictions as there are high chances of false alarms due to the number of false positive and false negative that can be present in the model. Building on these drawbacks, this research addresses the black box nature by integrating explainable AI techniques like LIME to enable clear feature-based explanations.

In the paper '*A Novel Deep Learning-Based Approach for Malware Detection*' by (Shaukat et al., 2023) a hybrid model which combines machine learning and deep learning techniques is proposed. The authors evaluated the performances of random forest and LSTM models to classify different malware types. Their findings show that hybrid models can leverage the strength of both machine learning and deep learning techniques, which achieve high accuracy, as well as it was able to differentiate among different malware families. However, they noted a major limitation, which was the computational complexity of hybrid models, which can be prohibited for real-time use as they are not very suitable in environments where computational resources are limited. To address the issue, this research highlights the importance of balancing advanced techniques with practical considerations, such as computational efficiency, for real-world implementation. So, the study aims to incorporate interpretable machine learning models that maintain a balance between computational efficiency and detection accuracy while addressing the transparency issues through explainable AI.

This paper by (Kimmell et al., 2021) '*Analyzing Machine Learning Approaches for Online Malware Detection in Cloud,* ' explores the effectiveness of the Support Vector Machine (SVM) for real-time malware detection. The authors conducted experiments to show that SVM is very effective in classifying malware with high accuracy if the features are properly selected. The study highlights the importance of feature engineering to optimise the model's performance. Even though SVMs are well-suited for malware classification, there is still the problem of black-box behaviour and also false predictions which can be a problem to trust the model completely. This research aims to address the issue by integrating the explainable AI technique LIME for making the predictions more transparent.

This research paper, '*A Novel Method for Malware Detection on ML-based Visualization Technique*' by (Liu et al., 2020) presents a comparison between different ML models, including Decision Trees, Gradient Boosting, and KNN. The research aimed to find an effective model for different malware datasets. The gradient boosting model was performing well with better accuracy, but it was less interpretable and expensive to implement. It was also harder to debug as well. The authors highlighted the importance of balancing accuracy and interpretability to ensure validated predictions are made by the model. This aligns with the research, which focuses on enhancing Interpretability through explainable AI.

The paper '*Malware Detection in IoT Devices Using Machine Learning: A Review*' by (Singh and Khurana, 2024) offers a broad overview of various machine-learning techniques used for malware detection in IoT environments. The authors compared supervised, unsupervised, and semi-supervised learning approaches for detecting malware and also highlighted the potential of each method type. The methods were performing well in detecting malware. The paper also discusses the importance of features importance and the challenges faced because of data imbalance. The authors have pointed out that although machine learning algorithms performed well in malware detection, model interpretability and resource requirements are critical factors that need to be considered to apply them in the real world. So, advanced methods must be brought up to tackle these issues. This aligns with the objectives of this research to address those critical challenges.

This paper ' *AI-based Malware and Ransomware Detection Models*' by (Marais et al., 2022) presents an AI-based approach for detecting malware and ransomware, especially focusing on Windows files. The authors proposed a hybrid model that combines both machine learning and deep learning techniques for malware detection and classification. This approach also identifies the type of malware that is detected. The authors used a combination of models such as LightGBM, XGBoost, DNN and CNN. The models performed well in terms of accuracy and classifying malware. But although they had good accuracy, there were difficulties in optimisation and also, similarly, the issues of black box behaviour since there are chances of false alarms as well. The future scope of this research paper also plans to incorporate Explainable AI to enhance the interpretability of the models. This aligns closely with the goals of this research to bridge the gap between accuracy and transparency in malware detection systems.

2.2 XAI Techniques for Malware Detection

Explainable AI is seen as a crucial enhancement to conventional machine learning models, Especially in the area of cyber security where trust accountability and transparency are given the most importance. Several studies have contributed to the development and use of XAI methods specifically for malware detection.

The paper titled '*Analyzing and Explaining Black-Box Models for Online Malware Detection*' by (Manthena et al., 2023) explores Local Interpretable Model-Agnostic Explanations (LIME) and Shapley Additive Explanations (SHAP) as post-hoc explanation techniques for machine learning models used for malware classification. The paper discusses the application of these methods to gain valuable insights into how features contribute to model predictions. The authors conclude that these techniques enhance the interpretability of machine learning models without significantly sacrificing accuracy. The LIME technique is known to generate local explanations quickly and is a preferred technique for malware detection. This aligns with the goals of this research, which also incorporates LIME to bridge the gap between accuracy and interpretability.

This paper by (Galli et al., 2024) '*Explainability in AI-based Behavioral Malware Detection Systems*', presents an approach for integrating global interpretability into malware detection models. The authors propose a method that combines Principal Component Analysis (PCA) with interpretable visualisations that show how different features impact the classification of malware samples. The study demonstrated that using PCA also enhances global interpretability, allowing security professionals to gain deeper insights into feature contributions. This paper emphasizes the importance of balancing interpretability with performance aligns with the goals of this research.

This research paper ' *Advancing Malware Detection using Memory Analysis and Explainable AI Approach*' by (Ravikumar et al., 2024) introduces a hybrid XAI framework that integrates both intrinsic explainability and post-hoc explanation. This paper explains that explainability serves as a valuable tool for cybersecurity experts for decision-making. The authors say that the decision tree model provides a good level of interpretability, which is very useful in understanding the fundamental decision rules. The research was shown to improve analyst trust and make real-time adjustments feasible.

The challenges faced by black-box models are presented in the paper '*An Explainable AI Approach for Android Malware Detection System Using Deep Learning*' by (Smmarwar et al., 2023), where the authors argue that the lack of transparency in black-box models limits their validity. They explain how XAI techniques like LIME can be employed to open up these black boxes which allows the analysts to validate the reliability of alerts generated by malware detection systems. This paper suggests that incorporating XAI methods and also improving their efficiency can be very useful in future research.

This paper ‘*A Comprehensive Investigation into Robust Malware Detection with Explainable AI*’ by (Baghirov, 2023) provides an explanation of how explainable AI techniques can be integrated into malware detection frameworks to address the black-box nature of the models. The authors focus on enhancing two XAI techniques: LIME and SHAP. By using this CICMalDroid dataset, the authors applied various ML models like LightGBM and Logistic Regression to detect malware while incorporating XAI techniques, which will provide valuable insights into the model’s predictions. Finally, the author says how important it is to balance detection accuracy with interpretability for malware detection systems in the field of cyber security.

This study ‘*Enhancing Malware Detection Through Convolutional Neural Networks and Explainable AI’* by (Mim et al., 2024) presents the XAI-AMD-DL model which is an Android Malware Detection (AMD) system that employs a hybrid architecture that combines CNN and Bi-Gated Recurrent Units (Bi-GRU). The author explains the importance of integrating explainable AI to provide transparency into how features influence malware detection. The dataset used was the CICAndMal2019 dataset which achieved an accuracy of 97.98%. The study also says that XAI techniques like SHAP could cause more computational costs and it will also be challenging for real-world applications as well.

The paper ‘*Explainable AI for Android Malware Detection: Towards Understanding Why the Models Perform So Well?*’ by (Yue et al., 2022) investigates the application of multiple XAI techniques, mainly focusing on how explainability can be used to improve malware detection. The authors evaluated the LIME technique to deep learning models to analyse their capabilities in providing meaningful explanations. The dataset used contains various classes of malware as well. The limitation of this study was the lack of computational resources for other XAI methods such as SHAP, whereas LIME was the preferred method as it was less expensive and also generated local explanations.

Many other studies highlighted where ML models performed with good accuracy but had limitations due to their black-box nature also in some research they have used the XAI technique, especially the LIME method but faced challenges due to computational costs and the problems in balancing accuracy with interpretability. This research aims to fill these gaps by integrating LIME with Machine Learning models to create a transparent and efficient malware detection framework.

Table 1: Different studies on malware detection

REFERENCE	DATASET	MODELS	BEST PERFORMING MODEL	BEST ACCURACY
Akhtar and Feng 2022	Custom	RFC, SVC, GBC, GNB, KNN, CNN	CNN	99.0%
Alomari et al., 2023	malimg	CNN, LSTM	CNN	97.49%
Shaukat et al., 2023	Custom	Hybrid (RFC+LSTM)	Hybrid model	97.8%
Kimmell et al., 2021	IoT Malware Dataset	SVM, Neural Networks	SVM	98.5%
Liu et al., 2020	Custom	Decision Tree, Gradient Boosting, KNN	Gradient Boosting	95.3%
Singh and Khurana, 2024	IoT malware dataset	Supervised, Unsupervised, Semi-supervised	Supervised	95.9%
Marais et al., 2022	CICMalDroid Dataset	LightGBM, XGBoost, CNN, DNN	LightGBM	98.2%
Manthena et al., 2023	Custom	-	-	96.7%
Galli et al., 2024	Custom	PCA, Neural Networks	PCA	96.5%
Ravikumar et al., 2024	Memory Analysis dataset	-	-	96.8%
Smmarwar et al., 2023	CICAndMal2019 Dataset	Hybrid DL model	Hybrid DL model	97.98%
Baghirov, 2023	Custom	-	-	94.0%
Mim et al., 2024	Malimg	CNN	CNN	97.49%
Yue et al., 2024	Android Malware Dataset	-	-	97.2%

3 Research Methodology

The main objective of this research is to detect malware and to develop an interpretable classification system by utilizing Explainable Artificial Intelligence. The dataset chosen for this research is a Malware Memory Dump dataset obtained from the online platform (Kaggle). This methodology involves detailed preprocessing of the dataset, training of the models, and application of interpretability techniques (XAI) to validate the model predictions. The research also aims to ensure a robust model evaluation and applicability by focusing on obfuscated malware threats (CyberCop, n.d). This section outlines the step-by-step process, from dataset analysis and preprocessing to model training, evaluation, and interpretation of results. The Figure below (Fig.1) is the workflow diagram for Malware Detection using Explainable AI:

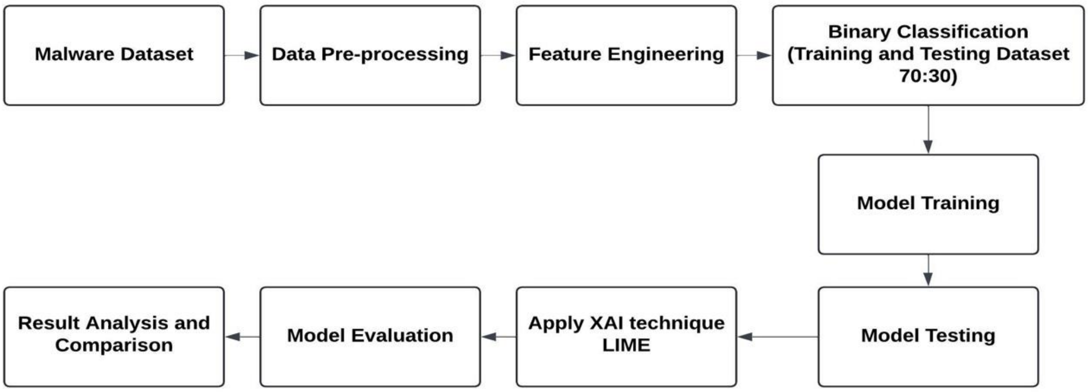


Figure 1: Workflow Diagram

This workflow represents the methodology for detecting malware using the memory dump dataset by integrating the XAI technique. It begins with collecting a balanced dataset of benign and malware samples. Then, the raw data undergoes preprocessing, such as handling missing values, simplifying categorical columns, encoding labels, and scaling numeric features to ensure consistency. Important features were selected for training, and also dimensionality was reduced, such as applying PCA for the SVM model.

After splitting the dataset into Training (70%) and Testing (30%) subsets, three machine learning models were selected- Random Forest, Support Vector Machine and Logistic Regression and they were trained and evaluated. Then, the XAI technique LIME is applied to interpret the model predictions, providing valuable feature-based explanations for each prediction. Furthermore, the model performance is analysed using metrics such as accuracy, precision, recall and F1-score, with results compared to determine the best-performing algorithm.

3.1 Dataset

The Dataset used for this research is a Malware Memory Dump dataset obtained from the online source Kaggle (CyberCop, n.d). This dataset consists of memory dump samples which were collected during system activities. The dataset has a balanced representation and comprehensive features which makes it helpful for training and evaluating malware detection models.

The memory dumps were collected in debug mode to ensure that no indications of the dumping process appeared in the files, maintaining their integrity and accuracy. The dataset consists of both benign and malware processes that are useful for a realistic scenario for testing detection systems.

Dataset Characteristics

The dataset contains 58,596 samples, which are equally distributed between benign and malware samples. Each sample is represented by 58 features describing various system-level attributes, such as process metrics, memory usage, and network activity.

Key features

The features are categorised as:

- System-level Attributes: Metrics such as number of processes, threads, and DLLs loaded.
- Network Activity: Features like bytes sent/received, packet counts, and connection duration.
- Categorical Attributes: Metadata such as ‘Raw_Type’ and ‘SubType’ describing the classification of memory dump.

Table 2: Key Features of Dataset

FEATURE NAME	TYPE	DESCRIPTION
pslist-nproc	Numeric	Number of processes running in memory.
pslist_threads	Numeric	Total thread associated with processes.
dlllist_ndlls	Numeric	Total number of DLLs loaded in memory.
handles_nhandles	Numeric	Total number of open handles in the system.
orig_bytes	Numeric	Bytes sent from the originating device.
resp_bytes	Numeric	Bytes received by the responding device.
duration	Numeric	Duration of the connection in seconds.
missed_bytes	Numeric	Number of bytes missed during transmission.
Raw_Type	Categorical	Original classification of the memory dump (Trojan, Spyware, Ransomware)
SubType	Categorical	Sub-classification of malware or benign activity.
Label	Categorical	Target column indicating benign or malware

Target Labels and Distribution

The target variable, Label, classifies the memory dump samples into two categories:

- 0 (Benign): No malicious activity was observed in the memory dump.
- 1 (Malware): Malicious activity detected, including obfuscated malware.

Table 3: Target Labels of Dataset

Label	Description	Count
0	Benign	29,298
1	Malware	29,298

The balanced distribution ensures unbiased model training and testing.

3.2 Dataset Preprocessing

To prepare the dataset for analysis, the following steps were applied:

1. **Simplification of Columns:** The ‘Raw_Type’ column was simplified to extract the main malware type, creating a new feature, ‘Raw_Type_Simplified’.

2. **Encoding of Labels:** The ‘Label’ was label-encoded to represent 0 as ‘Benign’ and 1 as ‘Malware’. And the ‘Raw_Type_Simplified’ feature was one-hot-encoded to generate binary columns for each malware type.

3. **Scaling:** Numeric features, such as ‘pslist_nproc’ and ‘orig_bytes’, were standardized using ‘StandardScaler’ to normalize their ranges.

4. **Feature Selection:** Irrelevant features, such as SubType, were removed to retain only the most relevant attributes.

5. **Handling Missing Values:** Missing values in columns like ‘missed_bytes’ were replaced with zeroes to ensure data completeness.

6. **Data Splitting:** The dataset was divided into 70% training and 30% testing subsets for unbiased model evaluation.

3.3 Model Implementation

Three binary classification models were trained to predict whether a memory dump sample is benign or malware:

1. **Random Forest:** An ensemble learning method based on decision trees and it is trained directly on the pre-processed dataset.

2. **Support Vector Machine (SVM):** Principal Component Analysis (PCA) was applied to reduce the feature space to 10 components and a liner kernel was used for classification.

3. **Logistic Regression:** Hyperparameter tuning was performed using GridSearchCV to identify the best parameters for model training.

3.4 Evaluation Metrics

The models were evaluated using the following metrics:

1. **Accuracy:** Proportion of correctly classified samples.
2. **Precision:** Ratio of true malware detection to all predicted malware cases.
3. **Recall:** Ratio of true malware detections to all actual malware samples.
4. **F1-Score:** Harmonic mean of precision and recall.

3.5 Explainable AI (XAI) Technique

LIME (Local Interpretable Model Agnostic Explanations

In this study, the LIME technique was used as a post-hoc interpretability technique to provide insights into the predictions made by the models. LIME is very effective in explaining complex, black-box models and making it easier for security professionals to understand the model's decisions.

How it works:

LIME works by perturbing the input data and observing the model's predictions to approximate the decision boundaries locally. It is able to create a simplified, interpretable model and give explanations about its prediction. The key features influencing the prediction are identified, ranked and visualized.

Importance of LIME in Malware Detection

Malware detection models such as Random Forest, Support Vector Machine and Logistic Regression, often operate as black-box models where the reasoning behind model predictions is not clear. So, LIME bridges this gap by:

- Generating local explanations for individual predictions
- Highlighting the contribution of specific features to classification decisions
- Provides insights into feature importance and model behaviour through visualizations.
- Building trust and interpretability

Application in this study:

In this research, LIME was applied to interpret the predictions of all three machine learning models- Random Forest, SVM and Logistic Regression. So, for each instance the LIME technique is used for:

- Feature importance scores were computed and visualised, which provided a detailed understanding of how individual features, such as ‘dlllist_ndlls’ and ‘handles_nhandles’ contributed to the classification decision.
- This process is helpful in understanding the strengths and limitations of the models, which ensures that predictions are interpretable and understandable.

3.6 Real-Time Monitoring and Alert System

To enhance the practical applicability of this research, a real-time monitoring and alert system was implemented. This system monitors the malware memory dump dataset and generates email alerts when any malware activity is detected. This feature is very useful as it is an additional layer of operational effectiveness, which allows security professionals to respond in time to potential threats.

Working Mechanism

- The system will continuously sample new data and process it through the ML model, and evaluates the probability of the sample being malware.
- If the model predicts a high probability of malware for example more than 70%, then the system triggers an alert.
- Based on the one-hot encoded features like ‘Raw_Type_Simplified_Ransomware’ or ‘Raw_Type_Simplified_Trojan’, the specific type of malware is identified.
- LIME technique is integrated into the models and explanations are generated for flagged instances which provides detailed insights into the features contributing to the prediction.
- Alerts are sent via email using third-party software (Mailgun API). However, the system can be extended to notify security professionals through additional means such as SMS alerts, Webhooks, and Dashboards for centralized monitoring.

Technical Implementation

- The system is designed with a customizable monitoring interval (10 seconds) to scan incoming memory dump data effectively.
- For email alerts, the Mailgun API software is used which helps in the transmission of detailed notification that contains the predicted class and probability, malware type and the LIME explanation of the prediction.
- The system can also be used with other notification channels based on the requirement.

The real-time monitoring alert system adds an additional layer of operational relevance to the study. It is ensured that the proposed methodology extends beyond predictive modelling by providing valuable insights through flexible alert mechanisms. This system can be customised as well according to requirements such as SMS, webhooks, etc., making it a good solution.

4 Design Specification

The foundation of this research lies in designing a strong architecture that uses machine learning and explainable AI techniques for malware detection. The malware memory dump dataset from (Kaggle) was used due to its rich representation of benign and malware samples, making it ideal for binary classification tasks. Data preprocessing steps, such as feature encoding, scaling, and dimensionality reduction, were implemented to prepare the dataset for model training.

Three machine-learning models were used to classify samples into benign or malware categories. To enhance interpretability, the LIME technique was integrated to provide feature-level insights into model predictions. In addition, a monitoring system with email alerts was added to extend the system's operation relevance. These components are discussed in detail in Sections 3 and 5.

5 Implementation

- The artefact was implemented using the Python programming language and various libraries. The malware memory dump dataset from the online platform (Kaggle) was preprocessed using the Pandas library. Key steps were simplifying the 'Raw_Type' column, one-hot encoding of categorical variables, and scaling numerical features.

- The dataset was split into training and testing sets in a 70:30 ratio. The 'Label' column was encoded into binary values, representing 'Malware' and 'Benign' classes, using Scikit-learn's 'LabelEncoder'.

- Dimensionality reduction was applied to the feature set for the Support Vector Machine (SVM) model using Principal Component Analysis (PCA). This reduced the computational complexity while retaining essential information for the classification tasks.

- Three models were implemented for binary classification tasks: Random Forest, SVM and Logistic Regression. The Scikit-learn library was used to develop the models.

- To enhance the system's interpretability, the LIME technique was integrated into the pipeline. LIME generated feature-level insights for individual predictions, showing the importance of specific attributes like 'pslist_nproc' (number of processes).

- A monitoring system was implemented to continuously evaluate incoming memory dump data. Predictions with a probability score above 70% for malware were detected and these predictions triggered the alert system to send email notifications via the software Mailgun API which included the predicted class, probability and explanations generated by the LIME technique as well.

- The outputs generated during the implementation included a properly preprocessed dataset, trained models evaluated with metrics, visualizations of LIME explanations for feature-level insights and an operational monitoring and alert system.

- The experiments were conducted on a MacBook Air M1, with an ARM64 processor and 8 GB RAM. The development environment used for the research project was Visual Studio Code integrated with Jupyter Notebook for iterative development.

6 Evaluation

This section provides a detailed analysis of the results obtained from the proposed system. The results are compared with the existing system, which shows the advancements of this research.

6.1 Hardware and Experimental Setup

The experiments were conducted on the following system:

- Device: Macbook Air M1
- Processor: ARM64
- Memory: 8 GB RAM
- Python Version: 3.10.5
- IDE: Visual Studio Code (v1.94.2)

6.2 Evaluation Metrics

To measure the performance of the proposed models, the following metrics were used:

- Accuracy: To measure the correct predictions made by the model. This metric measures the overall performance of the model.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

- Precision: This metric indicates the fraction of true positive predictions. It is critical for applications where minimizing false positives is essential, such as avoiding false alarms.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

- Recall: This metric shows the ability of the model to identify all true positives. It is important in malware detection to ensure no malicious software is missed.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

- F1-Score: It is the harmonic mean of precision and recall. It provides a balanced measure when precision and recall are equally important, especially where both false positives and false negatives must be minimized.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- Execution Time: This is the time taken for training and testing the model for checking efficiency.

Where TP is True Positive, and TN is True Negative. FP is False Positive, and FN is False Negative.

6.3 Model Performance

The proposed system evaluated Random Forest, Support Vector Machine (SVM) with PCA and Logistic Regression. The model Logistic Regression performed the best in accuracy when compared to the other two models as Random Forest showed overfitting issues and SVM took longer time for training and made false predictions.

- **Logistic Regression**

Table 4: Logistic Regression Model Performance

Accuracy	Precision	Recall	F1-Score	Training Time (s)	Testing Time (s)
0.994	0.993	0.994	0.994	3.2	2

- **Support Vector Machine**

Table 5: SVM Model Performance

Accuracy	Precision	Recall	F1-Score	Training Time (s)	Testing Time (s)
0.991	0.99	0.991	0.991	45	8

- **Random Forest**

Table 6: Random Forest Model Performance

Accuracy	Precision	Recall	F1-Score	Training Time (s)	Testing Time (s)
1.000	1.000	1.000	1.000	30	3

6.4 Comparison with Previous Study

The performance of the proposed system was compared with (Baghirov et al., 2024), which achieved an accuracy rate of 94% in one of the models they used. This comparison highlights the advancements in the proposed study in terms of metrics and practical application.

Table 7: Performance Comparison with Previous Study

Metric	Previous Study	Proposed Study
Accuracy	94%	99.4%
Precision	92%	99.3%
Recall	93%	99.4%
F1-Score	94%	99.4%

Analysis:

- The proposed study achieved a better accuracy, than previous studies.
- The proposed study enhanced the model’s interpretability by focusing on one XAI technique: LIME.
- The integration of an email alert system in the proposed study advances its practicality. Whereas, the previous study focused only on detection rather than utilising Explainable AI’s full potential by using it in real-time detection systems.

6.5 Explainable AI: LIME Analysis

To improve the interpretability and model prediction’s trustworthiness, LIME was used to explain individual predictions on the proposed study’s best model: Logistic Regression. This analysis shows the key features that influenced malware and benign classifications:

- ‘Raw_Type_Simplified_Spyware’ feature indicates whether the observed behaviour indicates malware type.

- ‘Raw_Type_Simplified_Trojan’ represents the presence of trojan-like behaviour.

- ‘Raw_Type_Simplified_Ransomware’ shows that there is a likelihood of ransomware-like activity.

- ‘callback_nanonymous’ captures the number of anonymous callback functions registered by processes, which could show malware-like behaviour.

- ‘svcscan_process_services’ measures the number of process-related services scanned during runtime. Anomalous numbers may indicate that there might be some malware activity.

- ‘svcscan_shared_process_services’ quantifies shared process service scanned. These could be exploited by malware for infiltration.

- ‘svcscan_nservices’ refers to the number of active services scanned on the system. High counts are expected to be signs of malicious tampering.

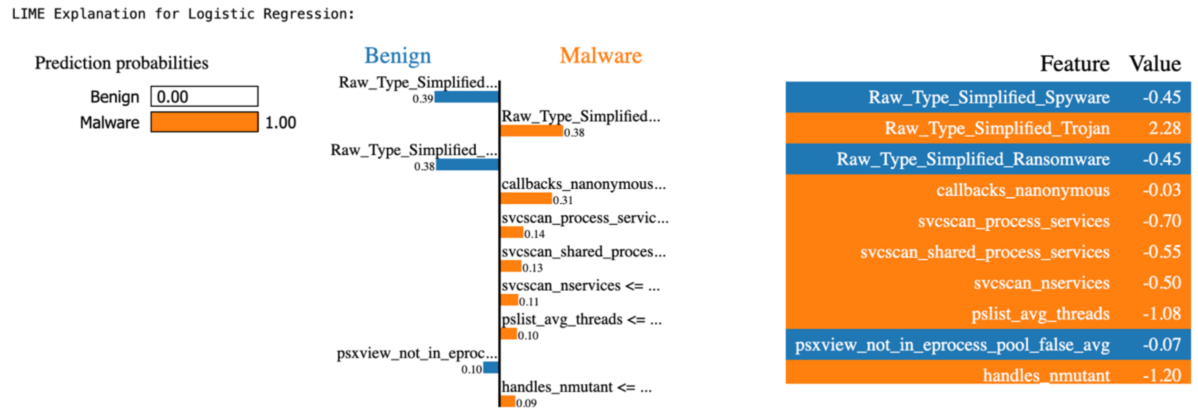
- ‘psxview_not_in_eprocess_pool’ checks for discrepancies in process visibility in the ‘process’ tool, this might indicate hidden processes.

- ‘handles_mutant’ measures the usage of mutant handles, which manages the synchronization in applications and abnormal usage can hint at signs of malware.

- ‘pslist_avg_threads’ captures an average number of threads per process in the system. A high thread count may indicate malware.

The visual explanations generated by LIME for both benign and malware predictions are shown below:

Malware Prediction:

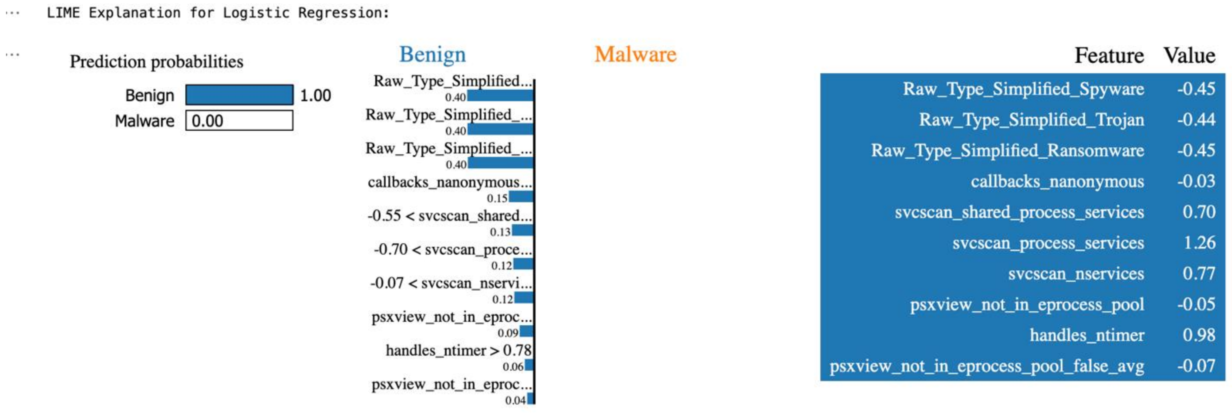


```
LIME Explanation for Logistic Regression:
```

Feature	Value
Raw_Type_Simplified_Spyware	-0.45
Raw_Type_Simplified_Trojan	2.28
Raw_Type_Simplified_Ransomware	-0.45
callbacks_nanonymous	-0.03
svcscan_process_services	-0.70
svcscan_shared_process_services	-0.55
svcscan_nservices	-0.50
pslist_avg_threads	-1.08
psxview_not_in_eprocess_pool_false_avg	-0.07
handles_nmutant	-1.20

Figure 2: LIME explanation for Malware class

Benign Prediction:



```
…  LIME Explanation for Logistic Regression:

…
```

Feature	Value
Raw_Type_Simplified_Spyware	-0.45
Raw_Type_Simplified_Trojan	-0.44
Raw_Type_Simplified_Ransomware	-0.45
callbacks_nanonymous	-0.03
svcscan_shared_process_services	0.70
svcscan_process_services	1.26
svcscan_nservices	0.77
psxview_not_in_eprocess_pool	-0.05
handles_ntimer	0.98
psxview_not_in_eprocess_pool_false_avg	-0.07

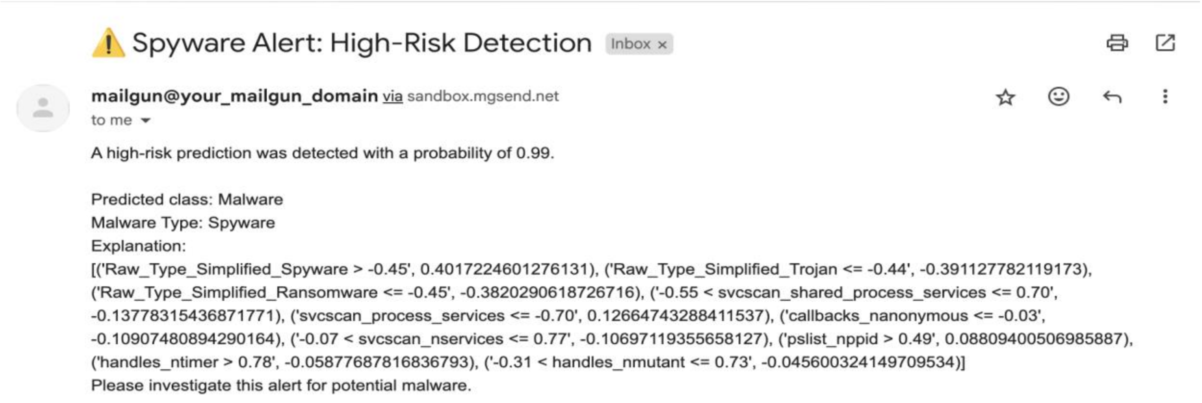
Figure 3: LIME explanation for Benign class

In these visualisations, the coloured bars represent the contribution of each feature to the model’s prediction. Positive values in orange bars strengthen the likelihood of the sample being malware, whereas negative values reduce the likelihood of the sample being malware.

Similarly, positive values in the blue bar indicate that features contribute to the sample being predicted as benign, but the negative features present in the blue bar indicate that it oppose the sample being classified as benign. To be precise, no matter what colour it is, the positive values support the prediction, whereas the negative values oppose the prediction. This interpretability allows us to understand how individual features influence the model's decision, thereby enhancing transparency and trust in the system's predictions.

6.5.1 Email Alert System

The implementation of an Email alert system ensures timely responses to malware detection. When a malware sample is detected with a probability exceeding 70%, an alert email is sent automatically to the specified administrator. This system uses a third-party software called Mailgun API for secure and efficient email delivery, including details such as predicted class, probability score, and summary of LIME explanations. The system can also be extended to include other notification mechanisms such as SMS etc.
This is what the email alert looks like:



⚠ Spyware Alert: High-Risk Detection

mailgun@your_mailgun_domain via sandbox.mgsend.net
to me

A high-risk prediction was detected with a probability of 0.99.

Predicted class: Malware
Malware Type: Spyware
Explanation:
[('Raw_Type_Simplified_Spyware > -0.45', 0.4017224601276131), ('Raw_Type_Simplified_Trojan <= -0.44', -0.391127782119173), ('Raw_Type_Simplified_Ransomware <= -0.45', -0.3820290618726716), ('-0.55 < svcscan_shared_process_services <= 0.70', -0.13778315436871771), ('svcscan_process_services <= -0.70', 0.12664743288411537), ('callbacks_nanonymous <= -0.03', -0.10907480894290164), ('-0.07 < svcscan_nservices <= 0.77', -0.10697119355658127), ('pslist_nppid > 0.49', 0.08809400506985887), ('handles_ntimer > 0.78', -0.05877687816836793), ('-0.31 < handles_nmutant <= 0.73', -0.045600324149709534)]
Please investigate this alert for potential malware.

Figure 4: Email Alert

6.6 Discussion

This study has demonstrated significant advancements in malware detection, combining accuracy and interpretability to address practical challenges in cybersecurity. Among the evaluated models, Logistic Regression emerged as the most reliable, achieving near-perfect scores in all metrics and outperforming previous studies such as Baghirov et al. (2023), which reported 94% accuracy. This improvement highlights the effectiveness of carefully tuned machine-learning models in malware detection tasks.

Another novel aspect of this research is the use of the XAI method, namely, LIME, to obtain feature-based explanations of model decisions. For example, features like 'svcscan_nservices' and 'handles_mutant' were recognized as critical signs of malware presence, increasing the level of transparency and, therefore, adding credibility to the system. These insights enable security professionals not only to identify the threats but also to understand the logic behind such threats, addressing the 'black box' portrayal of most machine learning-based predictions.

Further, the implementation of an email alert real-time system ensures quick response to malware predictions. The proposed system detects malware by setting the confidence threshold at 70%. The alert system also has potential for further scalability, for instance making use of other forms of notifications such as through the SMS or through a central monitoring dashboard thus improving operational usefulness.

However, there are several limitations: The Random Forest model was overfitting, so optimisation was needed. However, the dataset also lacks in terms of malware variants and size, which constrains the functionality of the system for a large number of classes of malware. To resolve these problems in the future work, the proposed framework will be reinforced.

In conclusion, it is seen that, besides technical gains, this study also emphasised practical applications by bridging the gap between accuracy, interpretability, and real-world usability. The findings and frameworks presented in this work serve as a solid basis for further enhancements in interpretable and scalable Malware Detection systems.

7 Conclusion and Future Work

This study addressed the problem of black-box problems faced in malware detection by using machine learning (ML) and explainable artificial intelligence techniques together by focusing on enhancing the accuracy as well as the interpretability of the ML models. The main objective was to explore various ML models, which were seen as black boxes due to their lack of interpretability, and integrate XAI techniques into them, such as LIME to provide transparency in decision-making and to improve the trustworthiness of model predictions. The study used a well-balanced malware memory dump dataset, and the methodology involved preprocessing, feature selection, and training and evaluating the models to find the best-performing model. And, finally adding an email alert system to send alert emails to the admin whenever there is a high probability that malware is detected. Although the models performed well, there were limitations, such as the overfitting issue in Random Forest and also the use of diverse datasets to strengthen the detection system as well.

Future Work

Although this study achieved the objectives, it has several ways for future research:

Real-Time Intrusion Detection Systems:
Another possible application of the developed models is deploying in real-time intrusion detection systems. These systems would actively scan networks so that when a new threat to the malware was discovered, it would be analysed instantly.

Ensemble Methods for Enhanced Performance:

Constructing hybrid models can help enhance the detection efficiency. For instance, integrating Random Forest with Logistic Regression or deep learning models could help derive the best from both to improve the accuracy as well as the robustness of the models.

Expanding Dataset Diversity:
Malware detection in this research work used memory dump data. Future studies should use datasets comprising of different types of malware such as APTs, and zero-day threats to ensure generalized results across different environments.

Integrating Multi-Source Datasets:
Extending the dataset to include network traffic, system logs, and memory dumps to get a more profound view of a malicious activity.

Advanced Explainable AI Techniques:
Although LIME served to offer localized model interpretation, extending XAI methods such as SHAP may prove valuable. These techniques would assist cybersecurity professionals in decisions made by a given model.

Exploration of Critical Domains:
The identified methodologies in this work could possibly be applied in other areas like the IoT and healthcare. Thus, the application of this system in detecting malicious activity in IoT networks or data security in healthcare systems could expand the system's scope.

This research successfully integrates machine learning and Explainable AI techniques for malware detection, which offer high accuracy and enhanced interpretability. By including a real-time email alert system and LIME explanations. The study ensures practical applicability as well as trust in model predictions. However, this system can be improved by using advanced models, and diverse datasets and extending it to critical domains like IoT and healthcare, ensuring a broader impact.

BIBLIOGRAPHY

European Union Agency for Cybersecurity (ENISA). (n.d.) Cyber Threats and Trends. Available at: https://www.enisa.europa.eu/topics/cyber-threats/threats-and-trends [Accessed: 28 November 2024]

Moore, T., Dynes, S., and Chang, F. (2018) 'Identifying how firms manage cybersecurity investment', *Computers & Security*, 77, pp. 65-76. Available at: https://doi.org/10.1016/j.cose.2018.04.007 [Accessed: 28 November 2024].

Capuano, N., Fenza, G., Loia, V., and Stanzione, C., 2022. Explainable Artificial Intelligence in CyberSecurity: A Survey. *IEEE Access*, 10, pp. 93575-93590. Available at: https://www.academia.edu/87662339/Explainable_Artificial_Intelligence_in_CyberSecurity_A_Survey [Accessed: 28 November 2024].

Ribeiro, M.T., Singh, S., and Guestrin, C., 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.1135-1144. Available at: https://dl.acm.org/doi/10.1145/2939672.2939778 [Accessed: 28 November 2024].

Akhtar, M.S. and Feng, T., 2022. Malware Analysis and Detection Using Machine Learning Algorithms. *Symmetry*, 14(11), p.2304. Available at: https://www.mdpi.com/2073-8994/14/11/2304 [Accessed: 28 November 2024].

Alomari, E.S., Nuiaa, R.R., Alyasseri, Z.A.A., Mohammed, H.J., Sani, N.S., Esa, M.I. and Musawi, B.A., 2023. Malware Detection Using Deep Learning and Correlation-Based Feature Selection. *Symmetry*, 15(1), p.123. Available at: https://www.mdpi.com/2073-8994/15/1/123 [Accessed: 28 November 2024].

Shaukat, K., Luo, S. and Varadharajan, V., 2023. A novel deep learning-based approach for malware detection. *Engineering Applications of Artificial Intelligence*, 122, p.106030. Available at: https://www.sciencedirect.com/science/article/pii/S0952197623002142 [Accessed: 28 November 2024].

Kimmell, J.C., Abdelsalam, M. and Gupta, M., 2021. Analyzing Machine Learning Approaches for Online Malware Detection in Cloud. *2021 IEEE International Conference on Smart Computing (SMARTCOMP)*, pp.189-195. Available at: https://ieeexplore.ieee.org/abstract/document/9556309 [Accessed: 29 November 2024].

Liu, X., Lin, Y. and Li, H., 2020. A novel method for malware detection on ML-based visualization technique. *Computers & Security*, 89, p.101682. Available at: https://www.sciencedirect.com/science/article/pii/S0167404818314627 [Accessed: 29 November 2024].

Singh, D. and Khurana, S., 2024. Malware Detection in IoT Devices Using Machine Learning: A Review. *2024 International Conference on Computational Intelligence and Computing Applications (ICCICA)*. Available at: https://ieeexplore.ieee.org/abstract/document/10585149 [Accessed: 29 November 2024]].

Marais, B., Quertier, T. and Morucci, S., 2022. AI-based Malware and Ransomware Detection Models. *arXiv preprint*. Available at: https://arxiv.org/abs/2207.02108 [Accessed: 29 November 2024].

Manthena, H., Kimmell, J.C., Abdelsalam, M., and Gupta, M., 2023. Analyzing and Explaining Black-Box Models for Online Malware Detection. *IEEE Access*, 11, pp.25237-25250. Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10064285 [Accessed: 29 November 2024]

Galli, A., La Gatta, V., Moscato, V., Postiglione, M., and Sperlì, G., 2024. Explainability in AI-based Behavioral Malware Detection Systems. *Computers & Security*, 141, p.103842.

Available at: https://www.sciencedirect.com/science/article/pii/S0167404824001433 [Accessed: 30 November 2024].

Ravikumar, C., Naresh, U., Manoranjini, J., Telang, S., Pallavi, S., and Kiran, S., 2024. Advancing Malware Detection Using Memory Analysis and Explainable AI Approach. *2024 Second International Conference on Intelligent Cyber Physical Systems and Internet of Things (ICoICI)*. Available at: https://ieeexplore.ieee.org/abstract/document/10696406 [Accessed: 30 November 2024].

Smmarwar, S.K., Gupta, G.P., and Kumar, S., 2023. XAI-AMD-DL: An Explainable AI Approach for Android Malware Detection System Using Deep Learning. *2023 IEEE World Conference on Applied Intelligence and Computing (AIC)*. Available at: https://ieeexplore.ieee.org/abstract/document/10263974 [Accessed: 30 November 2024].

Baghirov, E., 2023. A Comprehensive Investigation into Robust Malware Detection with Explainable AI. *Cyber Security and Applications*, 3, p.100072. Available at: https://www.sciencedirect.com/science/article/pii/S2772918424000389 [Accessed: 30 November 2024].

Mim, M.M.J., Nela, N.A., Das, T.R., Rahman, M.S., and Shibly, M.M.A., 2024. Enhancing Malware Detection Through Convolutional Neural Networks and Explainable AI. *2024 IEEE Region 10 Symposium (TENSYMP)*. Available at: https://ieeexplore.ieee.org/abstract/document/10752108 [Accessed: 30 November 2024]

Yue, L., Tantithamthavorn, C., Li, L., and Liu, Y., 2022. Explainable AI for Android Malware Detection: Towards Understanding Why the Models Perform So Well? *arXiv preprint*. Available at: https://arxiv.org/pdf/2209.00812 [Accessed: 30 November 2024].

CyberCop, (n.d.) *Malware Detection from Memory Dump Dataset*. Available at: https://www.kaggle.com/datasets/subhajournal/malware-detection-from-memory-dump [Accessed: 30 November 2024]).

Mailgun, (n.d.). *Email API - Send Reliable Transactional Emails*. Available at: https://www.mailgun.com/products/send/email-api/ [Accessed: 30 November 2024].