Enhancing Malware Detection through Machine Learning using XAI with SHAP Framework

Nihala Basheer [1], Bernardi Pranggono[1], Shareeful Islam[1,2], Spyridon Papastergiou[2,3] , Haralambos Mouratidis[4]

[1] School of Computing and Information Science
Anglia Ruskin University, East Road, Cambridge, UK
[2] Research and Innovation, MAGGIOLI S.P.A., Italy
[32]Department of Informatics, University of Piraeus, Greece
[4]Institute for Analytics and Data Science, University of Essex, UK
{nihala.basheer, bernardi.pranggono, shareeful.islam}@aru.ac.uk,
spyros.papastergiou@maggioli.gr, h.mouratidis@essex.ac.uk

Abstract. Malware represents a significant cyber threat that can potentially disrupt any activities within an organization. There is a need to devise effective proactive methods for malware detection, thereby minimizing the associated risks. However, this task is challenging due to the ever-growing volume of malware data and the continuously evolving techniques employed by malicious actors. In this context, machine learning models offer a promising approach to identify key malware features and facilitate accurate detection. Machine learning has proven to be effective in detecting malware and has recently gained widespread attention from both the academic and research sectors. Despite their effectiveness, current research on machine learning (ML) models for malware detection often lacks necessary explanations for the selection of key features. This opacity of ML models can complicate the understanding of the outputs, errors, and decision-making processes. To address this challenge, this research uses Explainable AI (XAI), particularly the SHAP framework, to enhance transparency and interpretability. By providing extensive insights into how each feature contributes to the model's conclusions, the approach further improves the model's accountability. An experiment was conducted to demonstrate the applicability of the proposed method, beginning with the training of the chosen machine learning models, including Random Forest, Adaboost, Support Vector Machine and Artificial Neural Network, for detecting malware, and concluding with the explanation of the decision-making process using XAI techniques. The results showed high accuracy in malware detection, along with comprehensive explanations of the feature contributions, which justifies the outputs produced by the models.

Keywords: Explainable Artificial Intelligence, Cyber Security, SHAP, Malware Detection, Artificial Neural Network, Random Forest

1 Introduction

In the ever-changing environment of digital security, the evolving nature of malware attacks poses a significant increase in sophisticate threats, which can disrupt the resilience of organizational business continuity. Recent data from Statista reveals a staggering 5.5 billion malware attacks in 2022 alone [1], with notorious incidents like WannaCry [2] underscoring the urgent need for advanced detection strategies. Both industry and research communities are actively engaged in the development of methods for malware detection, highlighting the pressing need to identify and manage potential digital infrastructure risks. Machine learning stands out as a promising approach for malware detection, particularly given the vast volume of malware data [3]. By enabling systems to learn from data, machine learning offers a proactive means to detect threats and continually adapt to evolving malware variants. However, the issue of explainability in malware detection using machine learning is a significant concern [4], particularly in the domain of cybersecurity where the stakes are high and the need for trust and transparency is paramount. Machine learning models, especially those based on complex algorithms like deep learning, can often function as "black boxes," providing little to no insight into how they arrive at their predictions [5]. This limited consideration for explainability can impact the informed decision-making and hampers with the wider adoption of ML in various critical security applications. Addressing this vital gap, our research focuses on understanding the decision-making process of ML models by adopting Explainable Artificial Intelligence (XAI) to interpret the model's outputs. This paper presents a novel malware detection method using a number of ML models and explains the results using the SHAP framework. XAI offers many advantages like improving the transparency and interpretability of the decision process. The feature importance analysis technique of the XAI can help in understanding the factors that affect the model's predictions and any possible errors. This kind of transparency can enhance the trust in the model's output.

This paper makes three main contributions. Firstly, it makes use of several machine learning models like Random Forest, Adaboost, Support Vector Machine (SVM) and Artificial Neural Network (ANN), in order to ensure thorough training. The performance of the models was evaluated to analyze their accuracy in malware detection. Secondly, the main component of the research, XAI with focus on SHAP was employed. SHAP makes effective use of cooperative game theory to credit the output of the models to their input features. The average marginal contribution of a feature, taking into account all potential combinations, is then computed to yield the Shapley value. These values are then utilized to gauge the local and global importance of features in the dataset. This will offer a clear understanding of how and why a model arrives at a specific result.Finally, an experiment was conducted to demonstrate the applicability and accuracy of the proposed model. The findings reveal that the models, bolstered by XAI insights, achieve high accuracy rates in malware detection. The integration of SHAP into our models not only sheds light on their internal mechanics but also establishes a stronger basis for trust in AI, underscoring its significance in advancing cybersecurity measures.

2 Related Work

2.1 Evolution of Malware

The evolution of malware has exhibited increased sophistication, complexity, and adaptability over time. In 1949, John Von Neumann introduced the concept of a self-replicating code sequence, laying the foundation for the earliest notion of a computer virus capable of autonomously generating updated versions [6]. Initially, malware's primary purpose was not malicious destruction or data theft but rather the highlighting of vulnerabilities in MS-DOS systems [7]. During this period, the damage caused by malware often resulted in brief system crashes due to excessive resource consumption. Notably, malware from this era was transparent to the user, often displaying visible messages or graphics on the monitor. The Creeper worm [8], created by Robert H. Thomas in 1971, though more of an experimental self-replicating program and not intended to be malicious, marked the emergence of new digital concerns [9]. Over time, malware evolved into more advanced forms, including trojans, spyware, and ransomware, often driven by financial motives. The development of polymorphic and metamorphic malware introduced the ability to alter code to avoid detection [10]. The rise of targeted attacks, like Advanced Persistent Threats (APTs), marked a shift towards highly sophisticated, state-sponsored cyber espionage. More recently, the Internet of Things (IoT) has expanded the attack surface, leading to the evolution of IoT-specific malware [11]. The integration of AI and machine learning in malware has also emerged, enhancing its ability to evade detection and optimize attack strategies.

2.2 Adoption of Machine Learning for Malware Detection

The rise in cyber threats has necessitated the development of advanced malware detection methods. In 2001, Schultz et al. [12] introduced a novel approach that combined data mining with machine learning for malware detection. Their study utilized a dataset of 4,266 programs, comprising both malicious (3,265) and clean (1,001) binaries. These binaries were sourced from various FTP sites and a Windows 98 system. The dataset was divided into training and test sets, with the former used to generate classifiers and the latter for performance evaluation. The classifiers were trained using static features extracted from binary profiles, including system resource information, strings, and byte sequences, without needing to execute the binary.

The paper by Firdausi et al. [13] addresses the inadequacy of traditional signature-based antivirus systems in detecting new and polymorphic malware. It proposes a combination of dynamic malware analysis and machine learning for more effective detection. The study assesses five classifiers, with the J48 Decision Tree showing the best performance. However, the paper's focus is limited to behavior-based techniques and does not include other malware detection methods. There is also a concern about the adaptability of their approach to rapidly evolving malware and the generalizability of their results, given the absence of multiple datasets or cross-validation.

Machine learning in malware detection, while promising [14], faces several notable limitations. One of the primary concerns is the vulnerability of models to adversarial attacks. Attackers can manipulate input data in subtle ways that cause these models to make errors, a significant risk in an area where adversaries are constantly devising new evasion techniques. Another challenge is the heavy reliance on substantial amounts of labeled data [15, 16]. Obtaining and labeling this data for training can be resource-intensive and sometimes impractical, especially when dealing with the vast and evolving landscape of malware.

2.3 Explainable AI for Malware Detection

Explainable AI methods are being used in malware detection to address the lack of transparency and understandability in black-box models. These methods aim to provide explanations for the decisions made by the models, increasing trust and accountability. Various techniques, such as XAI, attention maps, and explainability scores, have been proposed to achieve high detection accuracy and interpretability [17]. The unique hybrid strategy devised by Demertzis et al. [18], combines the Lipschitz constant and Shapley values to strengthen machine learning models against adversarial assaults while also improving explainability. The technique underscores the significance of global and local interpretability (GLI), shedding light on the decision-making processes and the intricate interactions among features within intelligent models. The paper by Kumar et al. [19] introduces XAISM-CTH, an XAI-based mechanism for cyber threat hunting, enhancing detection and understanding of threats while improving system security and performance. This model demonstrated a higher performance when assessed with the already existing models through practical implementation. Poddar et al. [20] proposed a two-stage stacked ensemble learning models that utilizes both, gradient boosting, and random forest. The models had a notable accuracy of 97% in detecting malicious URLs. They had also employed XAI to gain better perceptive of the model's decision-making process. It provided a clear justification of the impact of 21 features on the predictions that consists of different URL classes, including benign, phishing and malware. This method increased the model's interpretability and effectiveness.

In summary, malware continues to become more complex and has evolved over time. Existing works have contributed to the development of techniques for effective malware detection methods by using various machine learning models. However, there is a lack of focus on transparency and interpretability of the model output. As a consequence, the outcome of the model cannot effectively support the decision-making of models to tackle malware. This work contributes towards this direction by adoption of XAI to explain the model and support the decision-making.

3 Proposed Approach

The proposed method consists of three well-defined sequential steps, represented in Figure 1, that adopts XAI in the final step. This method, paired with various ML models, aids in visualizing model outcomes and clarifying the decision steps by spotlighting the most trusted and detailed traits. An overview of each of the steps is described in this section.

Step 1 (Data-Preprocessing): The initial step of the approach focuses on Data Preprocessing, which includes importing a publicly available dataset and then refining it using the Synthetic Minority Oversampling Technique (SMOTE) to correct the issue of class imbalances, thereby making the data fit for analysis. This step also involves a rigorous feature selection procedure that use the Extra Trees Classifier to identify prominent features, simplifying the dataset for subsequent analysis.

Step 2 (Malware Classification and Performance Evaluation using ML Models): Once the data has been preprocessed, the next step involves classifying the malware. The research utilizes four ML models including Random Forest, AdaBoost, SVM, and ANN for this purpose. These models were selected based on diverse strengths they have to offer. Random Forest can improve accuracy as it leverages ensemble techniques, Adaboost can improve performance by targeting the misclassified samples; and SVM and ANN can provide powerful techniques for different learning tasks. Together, these models allow the system to handle various malware forms and data attributes efficiently, improving its capacity to recognize and neutralize cybersecurity hazards**.**

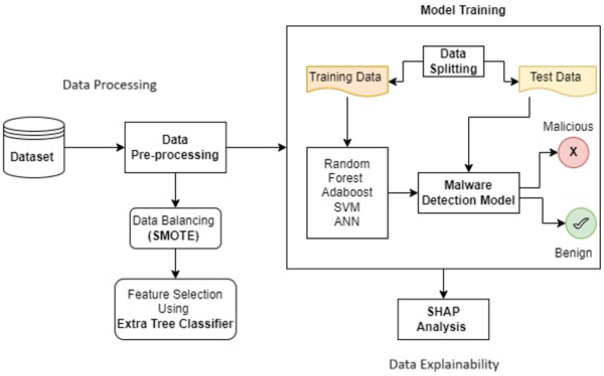


Fig. 1. Proposed Model

Step 3 (Interpretability of the models using XAI): The final step of the approach targets at Explainability, where the results from the trained machine learning models are not only evaluated but also interpreted using XAI techniques, especially the SHAP framework. SHAP (Shapley Additive Explanations) gives a clear understanding of how a model makes decisions. It calculates each input feature's contribution to the model's output predictions using a game theory mechanism known as Shapley values. This transparent approach helps to clarify the reasons behind the model's decision. These values assess each feature's contribution to the prediction by considering all possible feature combinations, offering a detailed insight into how each one influences the model's decisions. By integrating SHAP, the system goes beyond simple malware classification; it unveils the rationale behind these classifications. This transparency is crucial, as it demystifies the predictive mechanism, allowing users to grasp the significance and impact of each feature within the model's reasoning, thereby fostering trust and confidence in the system's predictions.

4 Experiment

This section details the implementation and goals of our methodology, leveraging a widely adopted dataset for this analysis. The experiment aims to:

- Detect malware using machine learning models and assess the accuracy.
- Demonstrate the impact of incorporating XAI and SHAP in enhancing the transparency and interpretability of malware detection processes.

4.1 Dataset Description

The "MalwareData.csv" dataset [21], has been used for model training. It consists of extracted static features from portable executable files. The dataset boasts a commendable size, with a total of 138,047. Notably, it encompasses 41,323 legitimate files, with the remaining samples being malicious. One of the salient features of this dataset is its comprehensive feature set, each mined through both static and dynamic analyses of software binaries. These features are curated to capture the intricate nuances and potential signatures that could differentiate benign software from malicious counterparts.

4.2 Data Pre-processing

Data pre-processing is a critical step in machine learning, ensuring consistent, unbiased, and effective model performance. The imbalance in dataset can pose a serious threat to the efficacy of the detection system as it could become more prone to overlooking harmful entities. To address this critical issue, researchers and practitioners often turn to the Synthetic Minority Over-sampling Technique (SMOTE). Unlike basic oversampling techniques that simply replicate minority samples, SMOTE generates synthetic data points by interpolating between existing minority samples in the feature space [22].

Feature selection is a critical process in machine learning that involves selecting a subset of the most informative features from the original dataset. By eliminating irrelevant or redundant features, the model not only becomes simpler, mitigating the risk of overfitting, but also often witnesses an improvement in accuracy, as superfluous features can sometimes misguide the learning algorithm. In this study, the feature selection phase was optimized using a method known as Extra Trees Classifier. This classifier, which is an ensemble technique, was trained on the dataset to identify the significance of each feature. Then, a filtering process was conducted to pick out only those features that had a certain level of importance.

4.3 Model Training

Random Forest, AdaBoost, SVM, and ANN are models, each with their own strength, that can be utilized to detect malwares. Random Forest is a technique that creates several decision trees from a dataset's subsets. By averaging the predictions of these individual trees, the technique improves the overall accuracy of the dataset predictions.

This approach is based on ensemble learning, where multiple classifiers work together to tackle a complex problem and boost model performance. AdaBoost, or Adaptive Boosting, is a supervised learning technique that classifies data by merging many weak or base learners into a strong learner. It emphasizes improving identification of challenging cases by adjusting weights for misclassified instances, increasing sensitivity to subtle malware indicators. SVM excels at classifying data in high-dimensional spaces, enabling effective differentiation between malware and benign software. ANN is a computational model that was inspired by the neural structure of the human brain. It comprises of linked nodes that are divided into layers. Information travels via these nodes, and the network changes the link weights during training to learn from data, allowing it to recognise patterns, and make predictions. These models together can improve the accuracy and precision for malware detection.

4.4 Model Evaluation

For the evaluation of the models, 5-fold cross validation was used to see how models perform on unseen data [23]. Important performance metrics such as accuracy, precision, recall, F1-measure, False Positive Ratio (FPR), and False Negative Ratio (FNR) were calculated [24,25]. These metrics play a pivotal role in evaluating the model's sensitivity-specificity equilibrium. Each fold assesses the model's performance independently, and the overall performance metrics are obtained by averaging across all experiments.

5 Experimental Result

5.1 Evaluation Metrics

Table 1 provides a comprehensive evaluation of four models. Each model's performance is assessed based on several key metrics. Among the classifiers evaluated, Random Forest, SVM and ANN exhibited the highest accuracy, achieving a score of 99%. It also boasted the lowest error rate among all classifiers, with just 0.01. Furthermore, Random Forest displayed the lowest False Positive Ratio (FPR) of 0.070 and a relatively low False Negative Ratio (FNR) of 0.031. AdaBoost followed closely behind Random Forest in terms of accuracy, with a score of 0.98. While AdaBoost had a slightly higher error rate of 1.38 compared to Random Forest, it maintained a commendable precision, recall, and F1-score. However, AdaBoost had a slightly higher False Positive Ratio (FPR) of 0.080 and a lower False Negative Ratio (FNR) of 0.013 compared to Random Forest. SVM and ANN both achieved an accuracy of 0.99, similar to Random Forest. However, SVM displayed a concerning performance with a False Positive Ratio (FPR) of 1.000, indicating that all negative samples were incorrectly classified as positive. On the other hand, ANN exhibited a lower False Positive Ratio (FPR) of 0.147 and a relatively higher False Negative Ratio (FNR) of 0.052 compared to the other classifiers.

Table 1. Performance of different classifiers and ensemble learning techniques

Classifiers	AC	ER	PR	RE	F1	FPR	FNR
Random Forest	0.99	0.01	0.99	0.99	0.99	0.070	0.031
AdaBoost	0.98	1.38	0.98	0.97	0.98	0.087	0.013
SVM	0.99	0.01	0.97	0.98	0.98	1.000	0.000
ANN	0.99	0.01	0.98	0.99	0.98	0.146	0.052

5.2 ROC Curve

Figure 2 depicts a Receiver Operating Characteristic (ROC) curve, assessing the performance of four models. The graph marks the true positive rate against the false positive rate, with the models significantly outperforming a random classifier, represented by the diagonal dashed line. The curves for Random Forest, AdaBoost, and ANN indicate perfect classification with AUC scores of 1.00, while the SVM shows a high AUC of 0.94. These high scores suggest excellent model performance.

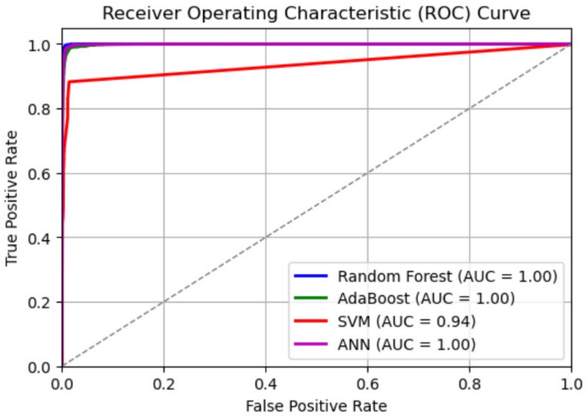


Fig. 2. ROC Curve of models

6 SHAP Analysis

SHAP analysis is a technique used in machine learning to explain the output of a model by attributing the prediction to the contribution of each feature. It provides insights into the importance and impact of individual features on the model's predictions. SHAP analysis helps in understanding the inner workings of complex machine learning models and provides interpretability by explaining why a model made a specific prediction. It mainly has two types of feature importance, global and local importance. Global feature importance provides an overview of the influence of each feature on the whole dataset. In contrast, local feature importance focuses on the contribution of each feature to the prediction of individual instances or observations. Combining global and local feature importance leads to a better knowledge of model behaviour. This can also further aid in model explanation, and trustworthiness in numerous applications. Overall, SHAP analysis enhances the interpretability of machine learning models and fosters trust and understanding among users and stakeholders.

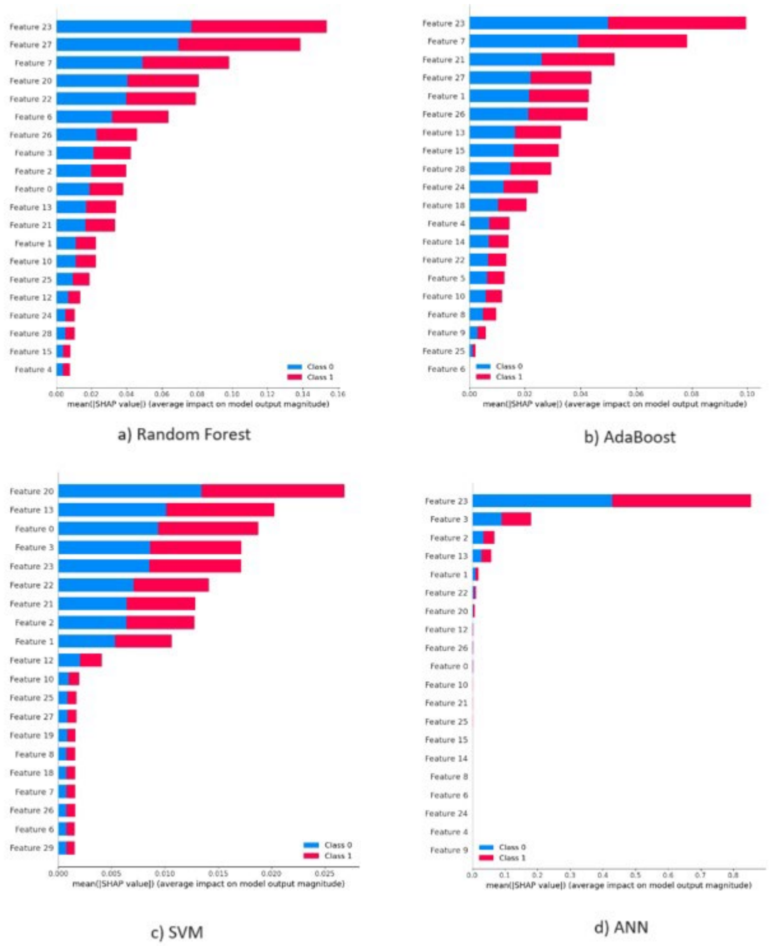
6.1 Global Importance of Features of Models

SHAP measures the contribution of each feature to the prediction of a particular instance relative to a baseline prediction, offering a way to interpret complex models – often referred to as the global importance of features. When visualized in a graph, SHAP values offer an intuitive display of each feature's impact: features are listed on the y-axis, and their corresponding SHAP values are on the x-axis. The color of the bars indicates the predicted class, and the length of the bar shows the magnitude of a feature's impact.

For the Random Forest depicted in Figure 3-a), 'SizeofHeaders' (Feature 23), 'SizeOfStackReserve' (Feature 27), 'SizeOfCode' (Feature 7) and ' MajorSubsystemVersion' (Feature 20) emerges as a significant predictor for both classes, with a substantial balanced impact on Class 0 and Class 1. However, 'Characteristics' (Feature 4) has the least influence on the model's output. Figure 3-b) corresponding to the AdaBoost model, again places 'SizeofHeaders' (Feature 23) at the forefront for both the class predictions. The distribution of feature impacts is slightly more balanced between positive and negative compared to the Random Forest model. The SVM model represented in Figure 3-c), we notice that the feature ' MajorSubsystemVersion' (Feature 20) is the most influential for Class 1 predictions, while 'SizeOfHeapReserve' (Feature 29) appears to have a negligible effect. This indicates a variation in how the SVM model interprets the features as compared to the Random Forest model. Lastly, the Artificial Neural Network model shown in the Figure 3-d) also indicates 'SizeofHeaders' (Feature 23) as a significant driver for Class 1 predictions, with a pronounced bias towards this class. The ANN model exhibits a different scale of SHAP values, pointing to a higher sensitivity to certain features compared to the other models.

6.2 Local Importance of Features of Models

The interpretability of machine learning models is a pivotal aspect of their development and deployment, particularly when decisions need to be understood and justified. Among the tools used for this purpose, force plots stand out as a powerful means of visualizing local feature importance. These plots illustrate how each feature value contributes to the overall prediction for a single data instance, compared to a baseline prediction. The force plot is akin to a tug-of-war, where features pull the prediction higher or lower, allowing us to see the relative influence of each feature within the context of a specific prediction.



a) Random Forest

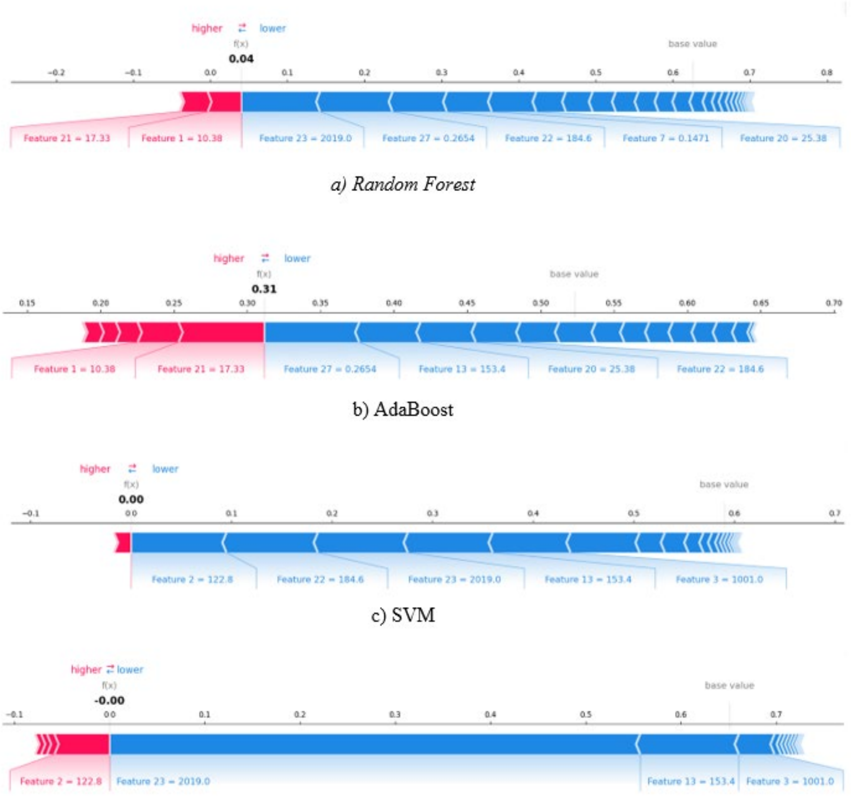
b) AdaBoost

c) SVM

d) ANN

Fig. 3. Cluster Plot Global Importance

Figure 4-a) corresponds to a Random Forest model's output. In this graphic, features depicted in red to the left are indicative of a negative influence on the model's prediction, while those in blue to the right positively contribute to the outcome. Notably, 'MinorSubsystemVersion' (Feature 21) and 'MD5' (Feature 1) exert substantial negative pressure, whereas 'SizeofHeaders' (Feature 23) 'SizeofImage' (Feature 22) and 'MajorSubsystemVersion' (Feature 20) are observed to have the most significant positive impact. Figure 4-b) corresponds to the AdaBoost model's output. This plot shares similarities with the Random Forest model in terms of the direction of influence exerted by the features; however, the magnitudes of their impacts vary, suggesting a differential feature importance in this model as compared to the Random Forest model. Features 'MD5' (Feature 1) and 'MinorSubsystemVersion' (Feature 21) are again seen to diminish the prediction score, while 'SizeofImage' (Feature 22), 'ImageBase' (Feature 13) and 'MajorSubsystemVersion' (Feature 20) elevate it. In Figure 4-c), SVM model, feature 'MajorLinkerVersion' (Feature 5) marginally detracts from the prediction, whereas features ‘SizeOfHeaders’ (Feature 23), ‘SizeOfOptionalHeader’(Feature 3) ‘SizeOfImage’ (Feature 22) and ‘ImageBase’ (Feature 13) augment it. The balance in the magnitude of these effects is more pronounced, highlighting a relatively equalized influence of features in this model, which is a contrast to the previous models. In Figure 4-d), for the ANN model, the features ‘Machine’ (Feature 2) and ‘SizeOfHeaders’ (Feature 23) have a diminishing effect on the prediction, suggesting a sway towards a negative class.



a) Random Forest

b) AdaBoost

c) SVM

d) ANN

Fig. 4. Force Plot for Local Importance

6.3 Summary of the SHAP Results

In summary, combining the results from both the global and local SHAP analysis, we can get an overview of the SHAP value interpretations for the four different models: Random Forest, AdaBoost, SVM, and ANN. However, the SHAP results cannot provide any evidence of prediction quality, but to address this issue, we have considered additional performance metrics such as accuracy, precision, recall, and the F1 score, alongside cross-validation scores to ensure robustness and reliability of our model predictions. These metrics allow us to quantitatively evaluate the models' performance and validate the predictive power of the features identified by the SHAP analysis. The insights we have inferred from the various plots are:

- Across models, certain features stand out as having consistently high importance for predictions, particularly for Class 1(Ref to Figure 3). The features 'SizeOfHeaders' (Feature 23) and 'SizeOfStackReserve' (Feature 27) are repeatedly significant for Class 1, i.e., Malware , predictions, although its impact varies across the models. For instance, SVM has higher contribution to the class 1 comparing to Random Forest. This indicates the importance of these features in malware detection. However, their impact varies when looking at the SVM model.
- The models seem to agree on the importance of several features, though the magnitude of their impact can differ, suggesting that while the features themselves are critical, their interpretation by each model can vary due to the intrinsic model complexities.
- The SHAP analyses enhance transparency by revealing which features drive model decisions. SHAP plots don't directly measure performance, they offer clues about model complexity and the potential trade-off between accuracy and interpretability. Features with large SHAP values may be key levers for predictive performance, yet they could also contribute to model complexity, affecting the ease of interpretation.

7. Conclusion

In response to the ever-evolving cyber threat landscape and the widespread adoption of malware for malicious activities, this research endeavors to advance the field of malware detection through a novel approach. Our proposal integrates Machine Learning with the XAI framework, specifically leveraging the SHAP methodology. This approach significantly enhances the explainability of ML models by offering a transparent view into their decision-making processes. This facilitates the detection of the key features and their contributions to the model's output. In doing so, it enhances the accuracy and fairness of the decision from the models while and limiting bias that could occur. The proposed approach significantly improves malware detection with accuracy rates above 98%. Additionally, the adoption of XAI enhances transparency, allowing for a better understanding of model decisions. It also fosters user trust by justifying ML decision-making processes. This kind of transparency is critical in the field of cybersecurity, where understanding the reasoning behind a model's prediction is as important as the prediction's accuracy itself. By using SHAP for XAI, cybersecurity professionals receive insight into not just the binary assessment of a file's maliciousness, but also the key factors that influence such decisions.

The proposed approach utilizes static malware detection, however it is necessary to adopt these models for real time malware detection which could offer a quicker and more reliable defense against malware attacks. It is also necessary to consider additional deep learning models for enhancing the quality of training and testing of the model. Finally, future works also needs to focus on assurance of Responsible AI properties to ensure that models are fair, reliable, unbiased and secure for informed decision making.

Acknowledgement
This work was supported by the European Union’s Horizon Europe Project, SEPTON-Security Protection Tools For Networked Medical Devices, funded under grant agreement No. 101094901 and CyberSecDome — An innovative Virtual Reality based intrusion detection, incident investigation and response approach for enhancing the resilience, security, privacy and accountability of complex and heterogeneous digital systems and infrastructures, funded under grant agreement No. 101120779.

References

1. Number of malware attacks per year 2022 | Statista. (2023, June 23). Statista. https://www.statista.com/statistics/873097/malware-attacks-per-year-worldwide/
2. B. Fiore, K. Ha, L. Huynh, J. Falcon, R. Vendiola, and Y. Li, "Security Analysis of Ransomware: A Deep Dive into WannaCry and Locky," 2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2023, pp. 285-294, doi: 10.1109/CCWC57344.2023.10099114.
3. Marko, Rebovich., Luka, M., Filipović., Ivana, Katnic., Milica, Vukotic., Tomo, Popovic. (2023). Machine learning models for statistical analysis. The International Arab Journal of Information Technology, doi: 10.34028/iajit/20/3a/8
4. Manthena, H., Kimmell, J. C., Abdelsalam, M., & Gupta, M. (2023). Analyzing and explaining Black-Box models for online malware detection. *IEEE Access*, *11*, 25237–25252. https://doi.org/10.1109/access.2023.3255176
5. Broll, B., & Grover, S. (2023). Beyond Black-Boxes: Teaching Complex Machine Learning Ideas through Scaffolded Interactive Activities. Proceedings of the AAAI Conference on Artificial Intelligence, 37(13), 15990–15998. https://doi.org/10.1609/aaai.v37i13.26898
6. Gaudesi, M., Marcelli, A., Sanchez, E., Squillero, G., & Tonda, A. (2016). Challenging antivirus through evolutionary malware obfuscation. In Lecture Notes in Computer Science. https://doi.org/10.1007/978-3-319-31153-1_11
7. Alenezi, M. N., Alabdulrazzaq, H., Alshaher, A. A., & Alkharang, M. M. (2020). Evolution of Malware Threats and Techniques: a Review. International Journal of Communication Networks and Information Security (IJCNIS), 12(3). https://doi.org/10.54039/ijcnis.v12i3.4723
8. Sahay, S. K., & Sharma, A. (2019). A survey on the detection of Windows Desktops malware. In Advances in intelligent systems and computing. https://doi.org/10.1007/978-981-13-5934-7_14
9. Yegneswaran, V., Barford, P., & Jha, S. (2004). Global intrusion detection in the DOMINO overlay System. Network and Distributed System Security Symposium. https://www.isoc.org/isoc/conferences/ndss/04/proceedings/Papers/Yegneswaran.pdf
10. Sharma, A., & Sahay, S. K. (2014). Evolution and Detection of Polymorphic and Metamorphic Malwares: A survey. *International Journal of Computer Applications*, *90*(2), 7–11. https://doi.org/10.5120/15544-4098
11. Shaukat, Ali., Omar, Abusabha., Farman, Ali., Muhammad, Imran., Tamer, AbuHmed. (2023). Effective Multitask Deep Learning for IoT Malware Detection and Identification

Using Behavioral Traffic Analysis. IEEE Transactions on Network and Service Management, doi: 10.1109/TNSM.2022.3200741

12. Bharadiya, J. P. (2023). Machine learning in cybersecurity: Techniques and challenges. European Journal of Technology, 7(2), 1–14. https://doi.org/10.47672/ejt.1486
13. M. G. Schultz, E. Eskin, F. Zadok and S. J. Stolfo, "Data mining methods for detection of new malicious executables," Proceedings 2001 IEEE Symposium on Security and Privacy. S&P 2001, Oakland, CA, USA, 2001, pp. 38-49, doi: 10.1109/SECPRI.2001.924286.
14. Konstantinos, Demertzis., Panayiotis, Kikiras., Nikos, Tziritas., Salvador, Llopis, Sanchez., Lazaros, Iliadis. (2018). The Next Generation Cognitive Security Operations Center: Network Flow Forensics Using Cybersecurity Intelligence. 2(4):35-. doi: 10.3390/BDCC2040035
15. I. Firdausi, C. lim, A. Erwin and A. S. Nugroho, "Analysis of Machine learning Techniques Used in Behavior-Based Malware Detection," 2010 Second International Conference on Advances in Computing, Control, and Telecommunication Technologies, Jakarta, Indonesia, 2010, pp. 201-203, doi: 10.1109/ACT.2010.33.
16. Konstantinos, Demertzis., Konstantinos, Demertzis., Lazaros, Iliadis., Elias, Pimenidis., Nikolaos, Tziritas., Maria, G., Koziri., Panagiotis, Kikiras., Michael, Tonkin. (2021). Federated Blockchained Supply Chain Management: A CyberSecurity and Privacy Framework. 627:769-779. doi: 10.1007/978-3-030-79150-6_60
17. Jo, J., Cho, J., & Moon, J. (2023). A malware detection and extraction method for the related information using the VIT attention mechanism on Android operating system. Applied Sciences, 13(11), 6839. https://doi.org/10.3390/app13116839
18. Konstantinos, Demertzis., Lazaros, Iliadis., Panagiotis, Kikiras. (2021). A Lipschitz - Shapley Explainable Defense Methodology Against Adversarial Attacks. 211-227. doi: 10.1007/978-3-030-79157-5_18
19. Kumar, P., Wazid, M., Singh, D. P., Singh, J., Das, A. K., Park, Y., & Rodrigues, J. J. P. C. (2023). Explainable artificial intelligence envisioned security mechanisms for cyber threat hunting. Security and Privacy. https://doi.org/10.1002/spy2.312
20. S. Poddar, D. Chowdhury, A. D. Dwivedi and R. R. Mukkamala, "Data Driven based Malicious URL Detection using Explainable AI," 2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Wuhan, China, 2022, pp. 1266-1272, doi: 10.1109/TrustCom56396.2022.00176.
21. PacktPublishing. (2018). Mastering-Machine-Learning-for-Penetration-Testing/Chapter03/MalwareData.csv.gz at master · PacktPublishing/Mastering-Machine-Learning-for-Penetration-Testing. GitHub. https://github.com/PacktPublishing/Mastering-Machine-Learning-for-Penetration-Testing/blob/master/Chapter03/MalwareData.csv.gz
22. Kerrie, Mengersen., Benoit, Liquet. (2023). SMOTE-CD: SMOTE for compositional data. PLOS ONE, doi: 10.1371/journal.pone.0287705
23. Warda, Aslam., Muhammad, Moazam, Fraz., S.K., Rizvi., Shahzad, Saleem. (2020). Cross-validation of machine learning algorithms for malware detection using static features of Windows portable executables: A Comparative Study.
24. Jude, Chukwura, Obi. (2023). A comparative study of several classification metrics and their performances on data. World Journal of Advanced Engineering Technology and Sciences, doi: 10.30574/wjaets.2023.8.1.0054
25. Islam, Shareeful, Abba, Abdulrazaq, Ismail, Umar, Mouratidis, Haralambos, Papastergiou, Spyridon(2022) ‘Vulnerability Prediction for Secure Healthcare Supply Chain Service Delivery’.Integrated Computer-Aided Engineering ,IOS Press.