**Predictive Modelling of Malware Behaviour using Advanced Machine Learning Algorithms**

**A Project Work Synopsis**

*Submitted in the partial fulfilment for the award of the degree of*

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE WITH SPECIALIZATION IN**

**INFORMATION SECURITY**

**&**

**ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING**

**Submitted by:**

21BCS3531 KIRTPREET KAUR

21BCS6405 AARUSHI

**Under the Supervision of:**

**Dr. Krishnendu Rarhi**



**CHANDIGARH UNIVERSITY, GHARUAN, MOHALI - 140413,**

**PUNJAB**

**April, 2024**

# Abstract

The research focuses on an innovative approach to use Random Forest, XG Boost and Decision Trees to enhance the malware prediction techniques. It helps to analyze large datasets containing behaviour logs and relevant information. By leveraging the ensemble capabilities of Random Forest, XG Boost, and Decision Tree, the system can effectively capture intricate patterns and relationships within the malware behaviour. Also, using Bagging Classifier enhances the robustness of the predictive model by aggregating multiple base learners and reducing over-fitting. By combining the strengths of machine learning models, the system can detect anomalies and identify potential malicious activity, thereby strengthening cyber security defences.

Keywords: Cyber Security, Decision Tree, Machine Learning, random Forest, XG Boost

# Table of Contents

|  |  |
| --- | --- |
| Title Page | i |
| Abstract | ii |
| 1.    Introduction |  |
| 1.1           Problem Definition |  |
| 1.2 Project Overview |  |
| 1.3 Hardware Specification |  |
| 1.4 Software Specification |  |
| 2.    Literature Survey |  |
| 2.1 Existing System |  |
| 2.2 Proposed System |  |
| 2.3 Literature Review Summary |  |
| 3.    Problem Formulation |  |
| 4.    Research Objective |  |
| 5.    Methodologies |  |
| 6.    Experimental Setup |  |
| 7.    Conclusion |  |
| 8.    Tentative Chapter Plan for the proposed work |  |
| 9.    Reference |  |

# 1. INTRODUCTION

## 1.1 Problem Definition

Develop a predictive model that can accurately identify and classify malware, including previously unknown variants, to enhance the security and integrity of computer systems and networks.

## 1.2 Problem Overview

Malware, including viruses, worms, trojans, and ransomware, poses a significant threat to individual and organizational computer systems, compromising data confidentiality, integrity, and availability. The rapid evolution and increasing sophistication of malware have rendered traditional signature-based detection methods inadequate. Therefore, it is essential to develop a predictive model that can identify and classify malware proactively, including previously unknown variants.

Goals:

1. Develop a machine learning model that can predict the likelihood of a file or program being malware.

2. Improve the accuracy and efficiency of malware detection and classification.

3. Enhance the security and integrity of computer systems and networks.

4. Reduce the risk of data breaches and cyber attacks.

Key Challenges:

1. High dimensionality and complexity of malware data.

2. Limited availability of labeled malware samples.

3. Rapid evolution and mutation of malware.

4. Need for real-time detection and response.

Success Metrics:

1. Accuracy and F1-score of malware detection and classification.

2. Reduction in false positives and false negatives.

3. Improvement in detection speed and response time.

4. Enhancement in overall system security and integrity.

By developing an effective malware prediction model, we can significantly enhance the security and integrity of computer systems and networks, protecting against the ever-evolving threat of malware.

## 1.3 Hardware Specification

1. Preferably 32 GB RAM (Minimum 16 GB)
2. At least 5 GB Storage Space
3. A stable Internet Connection with Minimum Bandwidth of 30Mbps

## 1.4 Software Specification

Python IDE (used Kaggle)

# 2. LITERATURE SURVEY

## 2.1 Existing System

The thesis explores malware detection using opcode analysis, introducing a new dataset and demonstrating the accuracy of this approach, particularly at short run lengths. It also shows the benefits of using assembly language descriptions over antivirus labels and achieves high accuracy in ransomware detection. The study evaluates the computational costs of various machine learning algorithms and identifies ensemble tree-based classifiers as top performers. A comprehensive analysis of classifiers across different phases is provided, with RandomForest(4) and RandomCommittee(4) emerging as the leading classifiers in terms of accuracy, training, and testing time [1]. The thesis addresses malware detection challenges by analyzing behavioral features and proposes a probabilistic approach to enhance cloud-based Forensic Virtual Machines (FVMs) scanning performance. It highlights the limitations of machine learning-based detectors and suggests a market- inspired prioritization method to balance resource consumption and accuracy. The approach integrates lightweight and heavyweight detection engines, achieving improved detection rates and reduced false alarms. The study also explores the use of machine learning algorithms for advanced malware detection, comparing static, dynamic, and hybrid analysis techniques and evaluating their effectiveness in detecting diverse malware families [2]. The research paper explores the use of machine learning algorithms for advanced malware detection, comparing static, dynamic, and hybrid analysis techniques. It evaluates the effectiveness of various algorithms, including Hidden Markov Models (HMMs), Support Vector Machines (SVMs), k-Nearest Neighbor (k-NN), Random Forests, and Convolutional Neural Networks (CNNs), in detecting diverse malware families. The study also investigates the use of clustering techniques, cryptanalytic techniques, and robust hashing for malware detection. The results show that dynamic approaches outperform static ones, and hybrid techniques are less effective due to feature incompatibility. The study concludes by highlighting the challenges in building a single model for diverse malware detection and recommends a combination of techniques for effective malware detection [3]. The thesis focuses on enhancing malware detector design, specifically for Android ransomware detection, by incorporating explainable machine learning techniques. It identifies key traits for effective detection, validates feature significance using explainability techniques, and explores the relationship between explainability and adversarial attacks. The study proposes a system that extracts System API information from Android applications, uses feature extraction strategies, and employs Random Forest classifiers for classification. The system is evaluated through experiments, demonstrating high accuracy, robustness against obfuscation, and resilience against adversarial attacks. Additionally, an Android application, R- PackDroid, is implemented for on-device detection, showing promising results in computational performance and feasibility on older devices. The study highlights the importance of explainability in assessing classifier resilience and demon- strates the effectiveness of the proposed approach in detecting Android ransomware [4]. This thesis proposes a deep neural network approach for static malware analysis, focusing on portable executable (PE) files. It aims to improve malware recognition accuracy and reduce false positives. The approach uses the EMBER dataset for training and achieves impressive results, with a 99.8% AUC and 98% true positives at 1% false positives. The model is designed to be practical and efficient, with potential for im- plementation as a complement or replacement for conventional anti-malware software. The thesis also discusses challenges in static malware analysis, such as signature avoidance and code obfuscation, and explores feature selection methods like opcode extraction and byte histograms [5]. The thesis explores the use of Machine Learning (ML) and Artificial Intelligence (AI) in malware analysis, highlight- ing challenges and limitations. It reviews the evolution of malware, types of malware, and detection models, including static, dynamic, and hybrid analysis. The study emphasizes the importance of combining static and dynamic features for ac- curate detection and classification. However, it also discusses limitations, such as false positives, lack of transparency and accountability, need for large datasets, and high computing re- quirements. The thesis concludes by emphasizing the potential of ML and AI in malware analysis while acknowledging the challenges that must be addressed for effective and reliable detection and classification [6]. The thesis proposes an innovative approach called adversar- ial reprogramming for few-shot malware detection, addressing the challenge of limited labeled data for training. The approach repurposes high-performance ImageNet classification models for malware detection, using features from malicious and benign files. Results show significant outperformance com- pared to baseline few-shot learning methods, demonstrating the effectiveness of the proposed approach against real-world malware. The thesis also explores the impact of pre-trained models, data sizes, and perturbation magnitude on the perfor- mance of adversarial reprogramming, highlighting its poten- tial for enhancing few-shot malware detection. Additionally, the thesis discusses the limitations and challenges of few- shot learning and malware detection, emphasizing the need for innovative approaches like adversarial reprogramming to improve cybersecurity [7]. AI in cybersecurity is crucial to combat evolving threats, with benefits like enhanced threat detection, automation, and improved response times. However, challenges like adversarial attacks and ethical considerations must be addressed through ongoing research, collaboration, and responsible implemen- tation to fully leverage AI’s potential in safeguarding the digital ecosystem [8]. The report warns about the emerging threat of AI- powered malware and phishing attacks, citing examples like DeepLocker and ChatGPT, which demonstrate the potential for AI to enhance cyber attacks. It highlights concerns about AI’s role in offensive cyber activities and emphasizes the need for ethical considerations, governance, and continued research into defenses against AI-generated malware [9]. The paper discusses the role of Artificial Intelligence (AI) in detecting malware, focusing on Machine Learning (ML) and Deep Learning (DL) techniques. It highlights the increas- ing sophistication of malware, the impact of the COVID-19 pandemic on internet usage, and the need for AI models that can detect various types of malware. The literature review covers recent research, datasets, analysis methods, and features relevant to AI-based malware detection. The paper concludes by outlining challenges and limitations faced by researchers, emphasizing the need for robust approaches to improve detec- tion rates against sophisticated malware [10]. The paper addresses the growing threat of ransomware attacks and presents a method for extracting encryption keys from memory, allowing for the decryption of encrypted files without paying a ransom. The approach focuses on the Salsa20 encryption algorithm, commonly used by recent crypto- ransomware strains. Through experimentation with real-world ransomware samples, the method successfully recovers over 90% of Salsa20 key and nonce pairs, enabling the decryp- tion of encrypted files [20]. The research demonstrates the effectiveness of live memory forensics in extracting encryption keys during ransomware execution, providing a means to recover files from advanced attacks without needing the master key. The findings highlight the potential for developing new mitigation techniques against cryptographic ransomware [11]. Malware is a big threat to digital world, and traditional methods to detect it are no longer working. New malware variants are getting smarter and can hide from detection. But, Deep Learning (DL) is helping us fight back! DL-based systems are better at finding new malware than old methods. They predict malware quickly and accurately, and even ana- lyze different types. This work looks at the latest DL-based malware detection systems and studies current malware trends, including mobile, Windows, IoT, APTs, and ransomware. It’s all about finding ways to stay safe in the digital world [12]! This research paper examines malware injection attacks, including techniques like XSS, SQL injection, and code injec- tion, and their consequences for targeted systems. It highlights the importance of countermeasures like input validation, se- cure coding practices, and web application firewalls to prevent and mitigate these attacks. The paper aims to provide valuable insights into malware injection attacks and effective defense strategies to enhance security posture and protect systems and sensitive information [13]. This paper analyzes the payload of various ransomware samples across different platforms (Windows, Android, Linux, and macOS) to identify trends and characteristics. The analysis covers 11 ransomware families, including WannaCry, Petya, and NotPetya, and proposes a set of 13 key characteristics to describe their behavior and design. The goal is to generalize the collected data and suggest threat mitigation techniques. The novelty of the paper lies in its analysis methodology, which helps determine similarities and differences among the ransomware samples. The authors manually analyzed the samples to eliminate contradictions in previous descriptions and verify the payload of the latest versions [14]. This paper explores Advanced Persistent Threats (APTs), a type of advanced malware that is highly sophisticated, target- specific, and operates stealthily until the target is compro- mised. APTs are capable of deploying automated malware and initiating on-demand attacks, using encrypted communication and advanced techniques that evade conventional security sys- tems. The paper discusses the limitations of traditional security measures and presents a detailed study on sophisticated attack and evasion techniques used by contemporary malware. It also discusses existing malware analysis techniques, application hardening techniques, and CPU-assisted application security schemes. Finally, the study concludes by presenting the Sys- tem and Network Security Design (SNSD) using existing mitigation techniques to counter APTs [15]. The paper aims to interpret the decisions made by machine learning models, specifically deep neural networks, in malware detection. It introduces a framework to analyze how the models generalize to unseen data and explains the mechanisms behind the MalConv architecture, a deep learning-based malware detection model. The analysis shows that the model assigns higher weights to specific portions of the executable, indicating their significance in classification. The framework allows for a better understanding of machine learning decisions and enables analysis of existing networks without retrainingg them [16]. The article addresses the growing concern of IoT malware propagation and introduces a new approach to modeling and mitigating its spread. The authors derive an exact Markov chain for random propagation of malware, which is validated through simulation results with high accuracy. Additionally, they propose a novel system of malware mitigation that operates at the network level, grouping and disconnecting

## 2.2 Proposed System

The proposed method focuses on both signature-based and behavior-based analysis for malware prediction, aiming to reduce the likelihood of malware entering the system.

For signature-based analysis, SHA-256 signature values are matched with Common Vulnerabilities and Exposures (CVE) values. Steps include loading the data, preprocessing, comparing CVE values with signature values, analyzing the results, and evaluating the performance using metrics like precision and recall.

For behavior-based analysis, key steps involve loading the data, removing duplicates and null values, generating descriptive statistics, checking correlations, selecting relevant features, transforming data if necessary, scaling features, splitting data into training and testing sets, encoding labels, selecting machine learning models (e.g., Random Forest, XGBoost), training each model, and evaluating their performance using metrics like accuracy and F1-score.

These methods together provide a comprehensive approach to predict and prevent malware intrusion.

## 2.3 Literature Review Summary

|  |  |  |
| --- | --- | --- |
| **Name of the Paper** | **Author Name** | **Algorithms Used** |
| Dynamic analyses of malware | Carlin, D. | Opcode analysis |
| Analysis and improvements of behaviour-based malware detection mechanisms | Alruhaily, N. | Machine learning and deep learning |
| Machine Learning Classification for Advanced Malware Detection | Fabio Di Troia | Machine learning algorithms |
| Malware Analysis and Detection with Explainable Machine Learning | Scalas, M. | RandomForest, RandomCommittee |
| Static Malware Detection Using Deep Neural Networks on Portable Executables | Puranik, P. | Deep neural networks |
| Machine Learning and Artificial Intelligence in Malware Analysis | Khan, T. W. | Machine learning algorithms |
| Few-Shot Malware Detection Using A Novel Adversarial Reprogramming Model | Ekula Praveen Kumar | Adversarial reprogramming |
| Artificial intelligence in cybersecurity: enhancing threat detection and mitigation | Mamadaliev, R. | machine learning algorithms |
| Artificial Intelligence Techniques for Malware Detection | AIRCC | machine learning algorithms |
| Exact Markov Chain of Random Propagation of Malware With Network-Level Mitigation | Carnier, R. M. et al. | Markov chain model |
| Federated Learning Approach for Distributed Ransomware Analysis | Vehabovic, A. et al. | Federated learning |
| Malware Detection and Prevention using Artificial Intelligence Techniques | Hossain Faruk, Md Jobair et al. | machine learning algorithms |
| Enviral: Fuzzing the Environment for Evasive Malware Analysis | Gorter, F. et al. | Fuzzing techniques |
| Mitigating the Risks of Malware Attacks with Deep Learning Techniques | Alnajim, A. M. et al. | Support Vector Machine (SVM), LSTM, CNN-LSTM |
| Trends in Malware Attacks: Identification and Mitigation Strategies | Pandey, A. et al. | machine learning algorithms |
| iOS mobile malware analysis: a state-of-the-art | Saudi, M. et al. | machine learning algorithms |
| Enhancing Cyber-Resilience for Small and Medium-Sized Organizations with Prescriptive Malware Analysis, Detection and Response | Ilca, L. F. et al. | machine learning algorithms |
| Artificial Intelligence Algorithms for Malware Detection in Android-Operated Mobile Devices | Alkahtani, H. et al. | Support Vector Machine (SVM), LSTM, CNN-LSTM |
| Explaining AI for Malware Detection: Analysis of Mechanisms of MalConv | Bose, S. et al. | MalConv |

.

# 3. PROBLEM FORMULATION

The problem involves developing a predictive model that can accurately classify a given file or program as malware or benign, including previously unknown variants. The goal is to learn a model that minimizes the error in classifying a given file or program, while handling high-dimensional feature vectors and generalizing well to unseen data. The model should also be able to detect malware in real-time or near real-time, making it a challenging optimization problem. The objective is to minimize the loss function, which measures the difference between the predicted and actual labels, subject to the model being in the set of possible models. By solving this problem, we aim to develop a predictive model that can enhance the security and integrity of computer systems and networks.

# 4. OBJECTIVES

1. Accuracy: Develop a model that can accurately classify a given file or program as malware or benign, with a high degree of precision and recall.

2. Generalizability: Train a model that can generalize well to unseen data, including previously unknown malware variants and new types of malware.

3. Robustness: Design a model that is robust to evasion techniques and can detect malware that tries to evade detection.

4. Interpretability: Develop a model that provides interpretable results, allowing for understanding of the reasons behind the classification.

5. Efficiency: Optimize the model for efficiency, to minimize computational resources and enable deployment in resource-constrained environments.

6. Scalability: Develop a model that can handle large volumes of data and scale to meet the needs of large organizations.

7. Flexibility: Design a model that can adapt to changing malware landscapes and new types of malware.

# 5. METHODOLOGY

The approach used in the research focuses more on MalwarePrediction than malware detection, thus, reducing the verylikelihood of malware entering the system. The analysis isbased on two methods:1. Signature-based analysis, which involved the use of patternmatching for strings of SHA-256 signature values.The steps involved in Signature-based malware analysisinclude:1. Load Data: Import the dataset containing the original signa-ture values and CVE (Common Vulnerabilities and Exposures)values.2. Preprocessing: This step involves cleaning and prepro-cessing the dataset if necessary, which may include handlingmissing values, removing duplicates, and ensuring data con-sistency.3. Comparison: Compare the CVE values with the originalsignature values. This comparison helps identify any matchesor discrepancies between the CVE values associated withknown vulnerabilities and the signature values present in thedataset.4. Analysis: Analyze the results of the comparison todetermine the extent of alignment between the CVE valuesand the original signature values. This analysis helps assessthe effectiveness of the signature-based approach in detectingknown vulnerabilities.5. Evaluation: Evaluate the performance of the signature-based analysis method based on the comparison results. Thisevaluation can involve calculating metrics such as precision,recall, and accuracy to measure the effectiveness of the ap-proach.Algorithm -1Step 1. // Load Datad a t a = l o a d d a t a s e t ( ’ d a t a s e t . c s v ’ )Step 2. // Preprocessingd a t a = p r e p r o c e s s d a t a ( d a t a )Step 3. // Comparisond e t e c t e d v u l n e r a b i l i t i e s = [ ]f o r e a c h c v e v a l u e i n d a t a :i f c v e v a l u e m a t c h e s any o r i g i n a ls i g n a t u r e v a l u e :r e c o r d d e t e c t e d v u l n e r a b i l i t y( c v e v a l u e )Step 4. // Analysist o t a l d e t e c t e d v u l n e r a b i l i t i e s= c o u n t ( d e t e c t e d v u l n e r a b i l i t i e s )p e r f o r m a n c e m e t r i c s= c a l c u l a t e p e r f o r m a n c e m e t r i c s( d e t e c t e d v u l n e r a b i l i t i e s )Step 5. // Evaluatione v a l u a t i o n r e s u l t s= e v a l u a t e p e r f o r m a n c e( d e t e c t e d v u l n e r a b i l i t i e s , known cves )Step 6. // End2. Behavior-based analysis, which included some key factorslikeThe steps involved in behavior-based malware analysisinclude:1. Data Load: Load the dataset into your programmingenvironment (Python, R, etc.).2. Sample Data: Optionally, take a sample of the data to workwith if the dataset is large.3. Duplicate Values Removal: Remove any duplicate rowsfrom the dataset.4. Null Values Removal: Remove or impute any null ormissing values in the dataset.5. Describe: Generate descriptive statistics for the dataset,such as mean, median, standard deviation, etc., to understandthe distribution of data.6. Correlation Check: Check for correlations betweenvariables to identify any redundant or highly correlatedfeatures.7. Selection of Columns: Select relevant columns or featuresfor your analysis and model building.8. Log Transformation: If necessary, perform a logtransformation on skewed data to make it more normallydistributed.9. Min-Max Scaler: Scale the data using Min-Max scaling tobring all features to a similar scale.10. Split Data Train and Test: Split the dataset into trainingand testing sets (e.g., 70% training, 30% testing).11. Label Encode: If your target variable is categorical,encode it into numerical labels.12. Model Selection: Choose the appropriate machine learningmodels for your task (e.g., Random Forest, XGBoost, DecisionTree, Bagging).13. Model Training: Train each selected model using thetraining data.14. Model Testing: Evaluate the performance of each modelusing the testing data, considering metrics like accuracy,precision, recall, F1-score, etc.Algorithm -2# Step 1: Data Loadd a t a = l o a d d a t a s e t ( ’ d a t a s e t . c s v ’ )# Step 2: Sample Datad a t a = d a t a . s a m p l e( f r a c = 0 . 5 , r a n d o m s t a t e = 1 )# Sample 50% o f t h e d a t a# Step 3: Duplicate Values Removald a t a = d a t a . d r o p d u p l i c a t e s ( )# Step 4: Null Values Removald a t a = d a t a . d r o p n a ( )# Step 5: Described e s c r i p t i o n = d a t a . d e s c r i b e ( )# Step 6: Correlation Checkc o r r e l a t i o n m a t r i x = d a t a . c o r r ( )# Step 7: Selection of Columnss e l e c t e d c o l u m n s =[ ’ column1 ’ , ’ column2 ’ , ’ column3 ’ . . . . . . ]d a t a = d a t a [ s e l e c t e d c o l u m n s ]# Step 8: Log Transformationd a t a [ ’ column1 ’ ] = np . l o g ( d a t a [ ’ column1 ’ ] )# Step 9: Min-Max Scalers c a l e r = MinMaxScaler ( )d a t a [ s e l e c t e d c o l u m n s ]= s c a l e r . f i t t r a n s f o r m( d a t a [ s e l e c t e d c o l u m n s ] )# Step 10: Split Data Train and TestX t r a i n , X t e s t , y t r a i n , y t e s t= t r a i n t e s t s p l i t( d a t a . d r o p ( ’ t a r g e t ’ , a x i s = 1 ) ,d a t a [ ’ t a r g e t ’ ] , t e s t s i z e = 0 . 3 ,r a n d o m s t a t e =4 2 )# Step 11: Label Encodel e = L a b e l E n c o d e r ( )y t r a i n = l e . f i t t r a n s f o r m ( y t r a i n )y t e s t = l e . t r a n s f o r m ( y t e s t )# Step 12: Model Selectionm o d e l s = [ R a n d o m F o r e s t C l a s s i f i e r ( ) ,X G B C l a s s i f i e r ( ) , D e c i s i o n T r e e C l a s s i f i e r ( ) ,B a g g i n g C l a s s i f i e r ( ) ]# Step 13: Model Trainingf o r model i n m o d e l s :model . f i t ( X t r a i n , y t r a i n )# Step 14: Model Testingr e s u l t s = {}f o r model i n m o d e l s :y p r e d = model . p r e d i c t ( X t e s t )a c c u r a c y = a c c u r a c y s c o r e( y t e s t , y p r e d )r e s u l t s [ t y p e ( model ) . name ]= a c c u r a c yp r i n t ( r e s u l t s )# Step 15: End

# 6.RESULT

The results of the analysis indicate that the signature-based approach achieved a notably high accuracy rate of98%. This method, relying on the comparison of CommonVulnerabilities and Exposures (CVE) values with the originalsignature values, demonstrated robust effectiveness in identi-fying and matching known patterns of malware behavior. Thebehavior-based analysis, exhibited a respectable accuracy rateof 85%, utilizing advanced machine learning algorithms suchas Random Forest, XGBoost, and Decision Trees followed bybagging Classifier

# 7.CONCLUSION

The research offers an innovative approach for predicting malware by utilizing the advantages of machine learning techniques like Random Forest, XGBoost, and Decision Trees. These techniques improve the prediction model’s accuracy and robustness when combined with the Bagging Classifier. The system can efficiently identify complex patterns and relationships inside malware behavior because of its capacity to evaluate big datasets that comprise behavior logs and pertinent data. This novel method greatly boosts cybersecurity defenses while also improving the detection of anomalies and possible hostile behavior. In order to further enhance the system’s prediction powers, future research could investigate the incorporation of additional machine learning models and the utilization of bigger, more varied datasets.

## 8. TENTATIVE CHAPTER PLAN FOR THE PROPOSED WORK

**CHAPTER 1: INTRODUCTION**

22nd February, 2024 to 29th February, 2024

**CHAPTER 2: LITERATURE REVIEW**

1st March, 2024 to 13 March, 2024

**CHAPTER 3: OBJECTIVE**

13th March, 2024 to 15th March, 2024

**CHAPTER 4: METHODOLOGIES**

16th March, 2024 to 31st March, 2024

**CHAPTER 5: CONCLUSION AND FUTURE SCOPE**

1st April, 2024 to 5th April, 2024

## REFERENCES

1. Carlin, D. (2018) Dynamic analyses of malware. Doctoral Thesis, Doctor of Philosophy, School of Electronics, Electrical Engineering and Computer Science, The Queen’s University of Belfast, N.Ireland.
2. Alruhaily, N. (2017) Analysis and improvements of behaviour-based malware detection mechanisms. Doctoral Thesis, Doctor of Philosophy, School of Computer Science, College of Engineering and Physical Sciences, The University of Birmingham.
3. Machine Learning Classification for Advanced Malware Detection by Fabio Di Troia May 2020, Doctoral Thesis, Kingston University
4. Scalas, M. (2021) Malware Analysis and Detection with Explainable Machine Learning. PhD Thesis, Università degli Studi di Cagliari, Cycle XXXIII.
5. Puranik, P. (2019). Static Malware Detection Using Deep Neural Networks on Portable Executables. Master's Thesis, University of Nevada, Las Vegas.
6. Khan, T. W. (2023). Machine Learning and Artificial Intelligence in Malware Analysis. Master's Thesis, University of Oslo.
7. Ekula Praveen Kumar (2022) Few-Shot Malware Detection Using A Novel Adversarial Reprogramming Model, Master's thesis, Wright State University
8. Mamadaliev, R. (2023). Artificial intelligence in cybersecurity: enhancing threat detection and mitigation. Scientific Collection InterConf, (157), 360–366. Retrieved from https://archive.interconf.center/index.php/conference-proceeding/article/view/3812.
9. HC3: Analyst Note. (2023, January 17). Artificial Intelligence and Its Current Potential to Aid in Malware Development. Analyst Note. TLP:CLEAR Report: 202301171500.
10. AIRCC. (2022, December). A Survey on Artificial Intelligence Techniques for Malware Detection. Academy & Industry Research Collaboration Center (AIRCC), 12(23).
11. R. M. Carnier, Y. Li, Y. Fujimoto, and J. Shikata, Exact Markov Chain of Random Propagation of Malware With Network-Level Mitigation, IEEE Internet of Things Journal, vol. 10, no. 12, pp. 10933-10947, 2023. doi: 10.1109/JIOT.2023.3240421.
12. Aldin Vehabovic, Hadi Zanddizari, Farook Shaikh, Nasir Ghani, Morteza Safaei Pour, Elias Bou-Harb, Jorge Crichigno. (2023). Federated Learning Approach for Distributed Ransomware Analysis. arXiv preprint arXiv:2306.14090 [cs.CR].
13. Hossain Faruk, Md Jobair, Hossain Shahriar, Maria Valero, Farhat Lamia Barsha, Shahriar Sobhan, Md Abdullah Khan, Michael Whitman, Alfredo Cuzzocrea, Dan Lo, Akond Rahman, Fan Wu. (2021). Malware Detection and Prevention using Artificial Intelligence Techniques. In 2021 IEEE International Conference on Big Data (Big Data). IEEE. DOI: 10.1109/BigData52589.2021.9671434
14. Floris Gorter, Cristiano Giuffrida, Erik van der Kouwe. (2023). Enviral: Fuzzing the Environment for Evasive Malware Analysis. In EUROSEC 2023. Association for Computing Machinery, Inc. DOI: 10.1145/3578357.3589455. Retrieved from
15. Abdullah M. Alnajim, Shabana Habib, Muhammad Islam, Rana Albelaihi, Abdulatif Alabdulatif. (2023). Mitigating the Risks of Malware Attacks with Deep Learning Techniques. Electronics, 12(14), 3166. DOI: 10.3390/electronics12143166. Retrieved from https://www.mdpi.com/2079-9292/12/14/3166.
16. Abhishek Pandey, Ashutosh Tripathi, Gayatri Kapil, Virendra Singh, Waris Khan, Alka Agrawal, Rajeev Kumar, Prof. Raees Khan. (2020). Trends in Malware Attacks: Identification and Mitigation Strategies. In Trends in Malware Attacks: Identification and Mitigation Strategies (pp. 47-60). doi: 10.4018/978-1-7998-1558-7.ch004. ISBN: 9781799815600.
17. Madihah Saudi, Muhammad Husainiamer, Azuan Ahmad, Mohd Idris. (2023). iOS mobile malware analysis: a state-of-the-art. Journal of Computer Virology and Hacking Techniques, 1-30. doi: 10.1007/s11416-023-00477
18. Lucian Florin Ilca, Petre Ogruţan Lucian, Titus Constantin Balan. (2023). Enhancing Cyber-Resilience for Small and Medium-Sized Organizations with Prescriptive Malware Analysis, Detection and Response. Sensors, 23(15), 6757. doi: 10
19. Hasan Alkahtani, Theyazn H. H. Aldhyani. (2022). Artificial Intelligence Algorithms for Malware Detection in Android-Operated Mobile Devices. Sensors, 22(6), 2268. doi: 10.3390/s22062268.
20. Shamik Bose, Timothy Barao, Xiuwen Liu. (2020). Explaining AI for Malware Detection: Analysis of Mechanisms of MalConv. In 2020 International Joint Conference on Neural Networks (IJCNN), 1-8. doi: 10.1109/IJCNN48605.2020.9207322.