Investigation of Machine Learning Algorithms for Malware Detection in PE and PDF Files

Masters in Cybersecurity

Rosemary Usoroh
Student ID: x21114374

School of Computing
National College of Ireland

Supervisor: Arghir-Nicolae Moldovan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Rosemary Uwem Usoroh
Student ID:	x21114374
Programme:	Master of Science in Cybersecurity **Year:** 2024
Module:	Msc Research Project
Supervisor:	Arghir-Nicolae Moldovan
Submission Due Date:	31-01-2024
Project Title:	Investigation of Machine Learning Algorithms for Malware Detection in PE and PDF Files
Word Count:	6619 **Page Count:** 24

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: Rosemary Usoroh

Date: 31-01-2024

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):	

Investigation of Machine Learning Algorithms for Malware Detection in PE and PDF Files

Rosemary Usoroh
x21114374

Abstract

Malware is a malicious program that uses harmful operations to destroy computer systems, get financial gain and steal confidential data. Many organizations lose their data, money and reputation because of malware attack. Therefore, malware detection is a crucial task in the cyber security field. Due to the dynamic nature of malware and the presence of new variants, the digital world must be protected from malware threats by the detection of malware using machine learning algorithms. Malware detection can be done in different file formats and files are the fundamental tools used to run software. The motivation of this research is to detect malware accurately in Portable Executable (PE) and Portable Document Format (PDF) files. This research contributes to the body of research by investigating the use of machine learning algorithms in the detection of malware. This work combined the use of four datasets with 33, 54, 92 and 631 features. Different machine learning (ML) algorithms were used to analyze the dataset. The machine learning algorithms includes, PART rule (PART), Ordinal Class Classifier (OCC), and Bayes Network (BN). The machine learning models were built and evaluated, the results from the experiments showed that OCC and PART models were the best classifiers with 100% accuracy on the WinMal dataset with 631 features. This research can be used for future work in malware detection and mitigation.

Keywords: Malware, detection, machine learning algorithms

1 Introduction

Malware attacks are a severe risk to individuals, organizations and the society (Kaur & Ramkumar, 2022). Cyberattacks can take many different forms, such as denial of service attacks, network invasions, phishing, and social engineering (Li & Liu, 2021). As we advance in technology and become more reliant on technology, the likelihood of malicious actors to take exploit vulnerabilities has increased as well (Trad et al., 2023).

PDF and PE files are one of the simplest methods for spreading malicious software. PDF and PE files have been used by cybercriminals for several cyber threats, such as phishing and the spread of malware (Yerima & Bashar, 2023). The malware can be distributed in different ways, such as email attachments and direct internet downloads.

This is presents a serious concern in cybersecurity because these files can be used to steal confidential data, destroy systems, and interfere with normal operations. The traditional methods of malware detection are unable to keep up with the latest new variants of malware

(Smith et al., 2023). To identify and stop malware attacks and safeguard our daily lives, businesses, industries, and healthcare, machine learning is a crucial tool (Faruk et al., 2022).

The primary objective of this paper is to identify the presence of malware accurately in a given PDF and PE file. This project aims to use machine learning algorithms on malware datasets for malware detection. The use of machine learning methods on the malware datasets will help mitigating malware threats. The research problem of the project motivates the following research question.

Research Question: How accurately can machine algorithms detect malware in PE and PDF files?

The major contributions of this paper are as follows;

- Investigates malware detection in PDF and PE files using machine learning algorithms.
- Provides a comparative study on malware datasets and machine learning algorithms.
- Use two different malware datasets to predict the model with high accuracy.
- Produce three machine learning models that will accurately detect malware.
- Evaluate the models metrics in terms of accuracy, precision and recall.
- Offer future direction of research in malware detection.

The structure of this work is organized as follows: Section 2 discusses related research on malware detection, Section 3 shows the research methodology, Section 4 shows the design specification of the research, Section 5 shows the implementation and Section 6 presents the evaluation and results of the research.

2 Related Work

Recently, malware has been targeting computers and organizations. When a user downloads a malicious email attachment or clicks on unsafe URLs, it begins to operate (Asaju et al., 2021). In order to address the issues of detecting malicious behaviour, various researchers have put forth various solutions. The use of machine learning algorithms to identify malware files has been the subject of numerous research studies to identify and stop malware attacks has also increased significantly (Almomani et al., 2021).

This literature review aims to provide insights into existing research on malware detection using supervised machine learning algorithms, focusing on datasets, results, and research gaps. A summary of previous research studies on the use of machine learning algorithms for malware detection is provided in this section.

2.1 Malware Datasets

Malware datasets are becoming more available these days and vast amount of malware samples have been collected and analyzed over the past few decades. It is easy to combat cyber attacks if we are aware of the functionality of malware datasets (Jeyalakshmi et al., 2022).

Researchers conduct their analysis using the publicly available datasets from different data repositories. Examples of dataset repositories are Kaggle, Google Dataset Search, VX Heavens, Microsoft research open data, University of California Irvine machine learning repository, Canadian Institute of Cybersecurity, GitHub, VirusShare, VirusTotal etc. Features in the dataset are the elements used in detecting malware.

The CWSandbox, created by ThreatTrack Security, and Anubis are used in some of the reviewed works. These sandboxes are mostly used to obtain malicious samples (Ucci et al., 2019). Also, Honeypots, Computer Emergency Response Teams (CERTs), and Internet service providers (ISPs) provide researchers with both benign and malicious datasets (Ucci et al., 2019).

EMBER, MOTIF, and RanSAP datasets do not permit access to original files for raw data-based analysis (Ramadhan et al., 2021). Although RanSAP only produces behavioural outputs, EMBER and MOTIF dataset only gives the static PE features and not the PE files. The Sorel-20M and DeepDetectNet (DDN) datasets offer binary file access for malware analysis based on raw data (Barut et al., 2023).

Malicia dataset consists of 11,688 binaries total from 500 drive-by download servers. Small features were analyzed with the dataset and dynamic features could not be used (Prajapati & Stamp, 2021). The datasets used in (Upchurch & Zhou, 2015) were selected based on the goal of malware detection. It had distinct groups of variants which contained varying numbers of samples.

Malimg dataset consist of 9,339 malware files from 25 different malware families, including worm, dialer, backdoor, Trojan, rogue and PWs. This dataset's problem is that there are no adversarial malware samples, the use of few hidden layers, and the lack of obfuscation techniques (Aslan & Yilmaz, 2021).

2.2 Malware Detection with Machine Learning Algorithms

Table I provides a summary of existing datasets and machine learning algorithms that were used in previous studies for the detection of malware. Most of the reviewed works used more than one machine learning algorithm to find out the one with more accurate results. These datasets and the various machine learning algorithms that were used produced different range of results.

Machine learning researchers find the malware detection problem ideal due to its large volume of data and advanced computing tools (Barut et al., 2020). Machine learning has been used for decades to solve the malware detection problem, and it seems to be the most logical solution (Barut et al., 2023). In malware analysis, the goal is to determine if a particular sample is malicious and it is very important because it enables for the prevention of a potential hazard in the system (Ucci et al., 2019).

(Mary et al., 2020) performed an analysis on malware detection. In this study, different malware and their intricate workings were examined. Samples of data were collected from the internet in different file formats. To get the intended results, they used different type of machine learning algorithms which includes Gradient Boosting Tree, Adaboost, Decision Tree, and Naïve Bayes. They also displayed a comparison of all the outcomes and the highest accuracy of 99.99% on the dataset. The actual source of their datasets was not dated and the raw files of the datasets were not made available for future comparison and analysis.

(Poudyal et al., 2018) used Random Forest, Decision Tree, and Navies Bayes with 97.95% accuracy for static analysis. They created a reverse engineering framework combining feature generation engines and machine learning (ML) to effectively detect malware. Researchers in (Issakhani et al., 2022) presented a stacking strategy that uses machine learning techniques to identify malicious PDF files. The extracted features from the dataset were used on the classifiers such as Support Vector Machine, Random Forest, Naïve Bayes, Gradient Boost, Ensemble Learning etc. The models performed well faster detected malware accurately.

Table I - Summary table for malware datasets and results

Citation	Malware Dataset	Malware Files Included Yes/No	Feature Processed Files Included Yes/No	ML Algorithm	AUC	Accuracy (%)	Train Split (%)	Test Split (%)	Class Values
(Herrera-Silva & Alvarez, 2023)	Cuckoo Sandbox	Yes	No	GNB GBT ANN RF	-	74.0 100 99.8 100	10f-CV	10f-CV	Malware/ Goodware Artifacts
(Poudyal et al., 2018)	TheZoo VirusTotal VirusShare	No	No	BN LR J48 RF	0.961 0.795 0.967 0.976	96.08 79.51 96.67 97.59	-	-	Malware/ Benign
(Akhtar & Feng, 2023)	Kaggle Library	Yes	No	RF DT KNN AB SGD EX GNB	-	1.0 0.99 0.99 0.99 1.0 1.0 1.0	10f-CV	10f-CV	Malware/ Benign
(Yousuf et al., 2023)	WinMal	Yes	Yes	NB SVM DT RF KNN NC GB	-	96.03 96.27 96.37 96.41 96.20 95.73 96.08	70	30	Goodware/ Malware
(Issakhani et al., 2022)	Evasive-PDFMal	Yes	Yes	RF MLP SVM AB	-	98.44 98.33 98.21 97.32	5f-CV	5f-CV	Malware/ Benign Files
(Kumar et al., 2019)	VirusShare	Yes	Yes	DT RF KNN LR LDA NB	-	74.24 74.24 79.55 73.48 81.82 28.79	10f-CV	10f-CV	Malware/ Goodware
(Tang et al., 2022)	Evasive-PDFMal	No	Yes	RF KNN LP DT SVM NB	-	96.97 94.33 77.55 96.89 95.03 96.41	60%	40%	Malware/ Benign
(Gusti & Girinoto, 2023)	Evasive-PDF Mal	No	Yes	GB XGB MLP NN	-	99.66 99.07 98.88 98.37	70%	30%	Malware/ Goodware
(Masum et al., 2022)	APT Malware	No	Yes	DT RF NN NB LR BN	0.99 0.99 0.99 0.73 0.99 0.98	0.98 0.99 0.97 0.35 0.89 0.97	70%	30%	Malware/ Legitimate Samples
(Yerima & Bashar, 2023)	Evasive-PDF Mal	Yes	Yes	AB ST RC RF	-	98.83 98.84 98.83 99.33	10f-CV	10f-CV	Malware/ Benign

Legend: Decision Tree (DT), Random Forest (RF), Naïve Bayes (NB), Logistic Regression (LR), Neural Network (NN), Gaussian Naive Bayes (GNB), Gradient Boosted Trees (GBT), Artificial Neural Networks (ANN), Bayesian Network (BN), AdaBoost (AB), Pre-Encryption dataset (PE), Pre-Encryption Detection Algorithm (PEDA), K-Nearest Neighbor(KNN), File Margin (FLM), LightGBM (LGBM), Voting Ensemble (VE), XGBoost (XGB), Decision Tree (J48), Linear Programming (LP), Support Vector Machine (SVM), Gradient Boost (GB), eXtreme Gradient Boosting (XGB), Multilayer Perceptron (MLP), Stacking (ST), Random Committee (RC).

(Anderson & Roth, 2018) used LightGBM, one of the gradient boosting decision tree models, to forecast the ember dataset and examine the outcome. The decision tree algorithm used in this dataset did not perform well, it had a 53% false positive rate and an 8% false-negative rate while LightGBM performed better with a AUC value of 99.911% and a false positive rate value of 92.99%. This study showed that the stale training dataset, dataset bias, or both contributed to the decision tree low performance compared to the LightGBM.

Researchers in (Yousuf et al., 2023) used a static malware detection technique that can accurately identify and categorise PE files malware in a Windows environment as either benign or malicious. The ML models include; K-Nearest Neigbour, Support Vector Machine, Nearest Centroid, Decision Tree Random Forest, Naïve Bayes, Gradient Boost and Enseble Learning methods.The static malware detection system The machine learning algorithms were used to classify the malware dataset and achieved a 99.5% detection rate.

More and new machine learning methods are being used to solve malware threats to mitigate and prevent the hazards to our society and cybersecurity as more and more malware malware data becomes available.

2.3 Research Gaps

Within the area of machine learning for malware detection, substantial advancements have been made. However, there are still several critical areas of research that require additional investigation. One crucial aspect to consider is feature selection, as previous studies highlighted its significance. The particular features that are most effective in detecting malware are still an area that requires further exploration.

Another important area of research is the availability of new datasets. Datasets that are older than five years should be regarded as obsolete (Barut et al., 2023). As new malware variants are increasing, new and recent datasets should be used for malware analysis.

The evaluation of current machine learning algorithms on evolving and new threats and understanding the adaptability of existing algorithms to counter evolving malware attacks is important for maintaining their effectiveness. This exploration would greatly improve the use of these ML algorithms in different scenarios.

This research complements previous studies by investigating the performance of several machine algorithms on different malware datasets. The reviewed literature discussed the advancements in detecting malware using supervised machine learning. It emphasized the importance of using different datasets and algorithmic approaches. The research gaps have highlighted the necessity for improved feature selection, use of new datasets, and the ability to adapt to new and evolving threats**.**

3 Research Methodology

The research methodology focuses on how the malware datasets were analyzed and how the models were built using different machine algorithms. This research involved a systematic approach following the Knowledge Discovery and Data Mining (KDD) methodology (Fayyad, 1996).This process involves five steps and it includes, data selection, preprocessing, transformation, data mining, evaluation and interpretation.

A. Dataset Selection and Description

In this research, four malware datasets were analyzed for this research. The first dataset was evasive PDF dataset (Evasive-PDFMal2022), it was downloaded from (Issakhani et al., 2022). The original file that was downloaded was in CSV format and it contained 10,027 rows and 33 columns. This PDF dataset consist of 10,025 PDF file samples (5557 malicious and 4468 benign) with no duplicate entries. It also had 37 extracted features (25 structural features and 12 general features) (Al-Taani et al., 2023).

This dataset was selected after going through past research papers as shown in Table I. It was selected because the dataset was recently developed by (Issakhani et al., 2022), it is well structured and it aligned with the objectives of the research. Table II shows the summary of original Evasive-PDFMal dataset and the feature description.

Table II – Summary Features Table for Evasive-PDFMal Dataset

Feature Name	Description
Pdfsize	The size of the pdf files in kilobytes
Metadatasize	The size of the metadata
Pages	Number of pages in the document
Title characters	Number of characters in the file title
isEncrypted	Whether or not the file is encrypted
Embedded files	Shows presence of embedded file
Images	Shows if the document contains images
Text	Shows document with text
Obj	Number of obj tags
EndObj	Number of endOj tags found
Stream	Number of stream tags
endstream	Number of endstream tags present
xref	Count of xref found
trailer	Number of trailers
startxref	Number of xref start indicator
/Page	Number of pages in PDF file
/Encrypt	Shows if the document needs a password
/ObjStm	Number of object streams
/JS	Number of JS objects
/JavaScript	Number of JavaScript objects
/AA	Automatic action
/OpenAction	Automatic action after viewing a document
/Acroform	Indicates traditional forms in Adobe Acrobat
/JBIG2Decode	Shows if JBIG2 compression is used
/RichMedia	Has embedded media
/Launch	Number of launch actions
/EmbeddedFile	Count of Embedded file keyword
/XFA	XML Forms keyword
/Colors	Shows the number of colours present

Table III – Summary table for EvasivePDFMal dataset attributes

Data File	No of Records	Description	Attributes	No of Attributes
EvasivePDFMal	10027	Contains several PDF file samples	Filename, pdfsize, metadata, pages, xreflength,titlecharacter, images, embedded	37

The second malware dataset which is Windows PE Malware (WinMal) samples was downloaded from (Yousuf et al., 2023). The samples were divided into four different parts which includes Imported DLLs, API Functions, PE Header and PE Section. The original files of these samples were all downloaded in CSV format. Each malware family's imported DLLs are listed in the first feature set (DLLs_Imported.csv file).

Table IV – Summary table for WinMal files dataset

Data File	No of Records	Description	Attributes	No of Attributes
DLLs_Imported.csv	29499	Contains names of imported DLLs.	SHA256, Type, advapi32.dll, Kernell32dll, vspmsg.dll, ole32.dll, oleaut32.dll, psapi.dll	631
API_Functions.csv	22537	Values of the API functions that is called by the malware	Getprocaddress, corexmain,exitprocess, loadlibraya getlasterror, getcurrentprocess,sleep	16384
PE_Header.csv	29808	Contains values of the PE Header files	SHA256, Type, e_magic, e_cblb, e_cp, e_crlc, e_cparhdr, e_minalloc, e_maxalloc	54
PE_Section.csv	29761	Contains values of the PE_Section.	SHA256, Type, text_misc_virtualsize, text_virtualaddress, text_sizeOfRawData, text_PointerToRawData	92

Table V – Summary Features Table for WinMal Files Dataset

Feature Name	Description
SHA256	The bit size of the hash values.
Type	The type of malware family
e_magic	Numerical value to identify the header files
NumberofSections	Number of sections in the header file
TimeDateStamp	The current date and time in the header
NumberOfSymbols	Number of symbols in the header file
ImageBase	Shows the file contains images at the base
SizeOfImages	Size of the images in the header file
Characteristics	Flags that shows the characteristics of the header
AddressOfEntryPoint	Memory address where the program is executed
Name	8-byte encoded string with name of the section
Misc_VirtualSize	Size of the section when in memory
SizeofRawData	The section’s size
Virtual Address	Address of the section’s first byte
NumberofRelocations	Count of relocation entries in the section
NumberofLineNumbers	Count of line number entries in the section
Characteristics	Flags that shows the characteristics of the section
.text	Shows program entry point
comdlg32.dll	Performs dialog box functions in windows
Kernel32.dll	Handles memory usage in windows
Vspmsg.dll	Repository for data, code and resources
Advapi32.dll	Provides access to extra functionality
Psapi.dll	Helps to get information about processes
Ole32.dll	Object link and embedding library
Oleaut32.dll	Helps programs to share data
Setupapi.dll	General setup and device installation functions

The SHA256 values are listed in the first column, the malware's label is listed in the second, and the names of imported DLLs are listed in the other columns. The second feature set (API_Functions.csv files) includes the malware labels and SHA256 hash values of the API functions that these malware calls. The third feature set (PE_Header.csv file) has values of the 52 PE header fields and the fourth feature set (PE_Section.csv file) has values from ten distinct PE sections (Yousuf et al., 2023).

The WinMal files dataset was selected because it is a new dataset, it has only been used in one previous study (Yousuf et al., 2023) and the dataset has different parts of PE files in it which is also very useful. Table III, IV and V presents the summary of Windows PE Malware dataset and feature description.

B. Data Preprocessing and Data Transformation

Large volumes of data are needed for machine learning algorithms to train the model before they can produce good results. Before the data is sent to the machine learning model for training, we must preprocess it to get better results. Data pre-processing includes checking for missing values and removing any data that will cause issues (Shroff & Maheta, 2015). From the author’s papers (Issakhani et al., 2022) (Yousuf et al., 2023), the features in the dataset were identified and their structure was easy to understand.

The Evasive PDFMal2022 and WinMal datasets came in a structured format, therefore needs little preprocessing. In the Evasive PDFMal2022, the row with NaN was removed on the excel sheet so that it is not treated as a separate class and for the classifier to be able to use the features. Inaccurate and corrupted data were removed from the dataset on the excel sheet.

In the WinMal dataset, the second feature set (API_Functions.csv files) was not included in the analysis. This is because the size of the dataset was too large (1.21 GB size and 16384 features) and the WEKA tool will not be able to build models on it. The other datasets which are DLLs_Imported.csv file, PE_Header.csv file and PE_Section.csv file were used for the malware analysis. The SHA column in the datasets was removed in WEKA to get accurate results from the classifiers.

C. Data Mining

In this step, the model will be defined, trained, tested, and used to make predictions. The goal of this research is to evaluate how well the classifiers predict malware and the accuracy of the result. The machine learning algorithms that were used for the data modeling includes; PART Rule, Ordinal Class Classifier and Bayes Network.

PART Rule: The attribute PART rule-based classifier is used to calculate the percentage of instances that the created rules should cover. It is a parameter that manages the trade-off between the rules' accuracy and understandability. The rule-based classifier employs the PART algorithm, which seeks to identify rules that cover a significant portion of the data while retaining a high degree of accuracy.

The degree of coverage necessary for a rule to be deemed valid by modifying the part attribute was regulated. This machine learning algorithm was selected because the performance and accuracy of the machine learning model is good (Al-Taani et al., 2023).

Ordinal Class Classifier: Ordinal class classification (OCC) is a specific type of multi-class classification model in which the instances are labelled by ordinal scales, and the output variables follow a natural total ordering. (Shi et al., 2019) confirmed that ordering information between labels in datasets helps to create more accurate models.

Ordinal classification has various applications since there is often an ordered relationship between the classes in real-world scenarios. This machine learning algorithm was selected because it enables standard classification of the malware datasets by using ordering method (He, 2022) and it has not been used on any of the malware datasets.

Bayes Net: The ability of a Bayes Net to predict the likelihood of observing a particular data pattern in the event that a given hypothesis is true sets it apart from other machine learning models. A probabilistic generative model is called a Bayes Net (Poudyal et al., 2018). A generative approach begins with a hypothesis and assumes prior knowledge.

This machine learning algorithm was selected because data can be analyzed by the machine learning model to identify links and trends, as well as to provide data-driven methods prediction-making process.

D. Evaluation

The evaluation of the model was carried out on the malware datasets. 10-fold cross validation was used on both datasets. This approach was selected based on previous research papers (Yerima & Bashar, 2023) that used 10-fold cross validation on malware datasets. The following metrics were used to evaluate the performance of the machine learning models;

Accuracy: This is the ratio of the total number of correct predictions and total number of predictions. This is the main goal of this research, to accurately detect malware in PE and PDF files.

Precision: This is the number of times a prediction is correct. This metric is useful when we have an unbalanced dataset and where there is large number of false negatives. Recall is a good metric to use in this research.

Recall: This helps in evaluating how good the models are in detecting malware files accurately. It calculates how many positives the models were able to capture.

F1 Score: This is a function of precision and recall. It strikes a balance between both metrics. All these metrics were chosen because they were used in previous research works (Al-Taani et al., 2023) and they help in the detection of malware in PE and PDF files

4 Design Specification

The research methodology employed a comprehensive approach, incorporating a range of machine learning algorithms to enhance the precision of malware detection. The collection consisted of different algorithms, each bringing their own strengths to enhance the overall effectiveness.

The use of PART rule model, a decision tree algorithm, offered valuable insights into the decision-making process that drives the model. Bayes Net, with its utilization of probabilistic graphical models, demonstrated its ability to effectively capture relationships between variables. Ordinal Class Classifier, an ordinal classification algorithm, demonstrated its efficiency in handling large datasets. This comprehensive integration of diverse machine learning algorithms formed the basis for an effective approach to addressing malware threats.

Figure 1 below shows a visual representation of the machine algorithm models from the data collection stage to the evaluation stage.

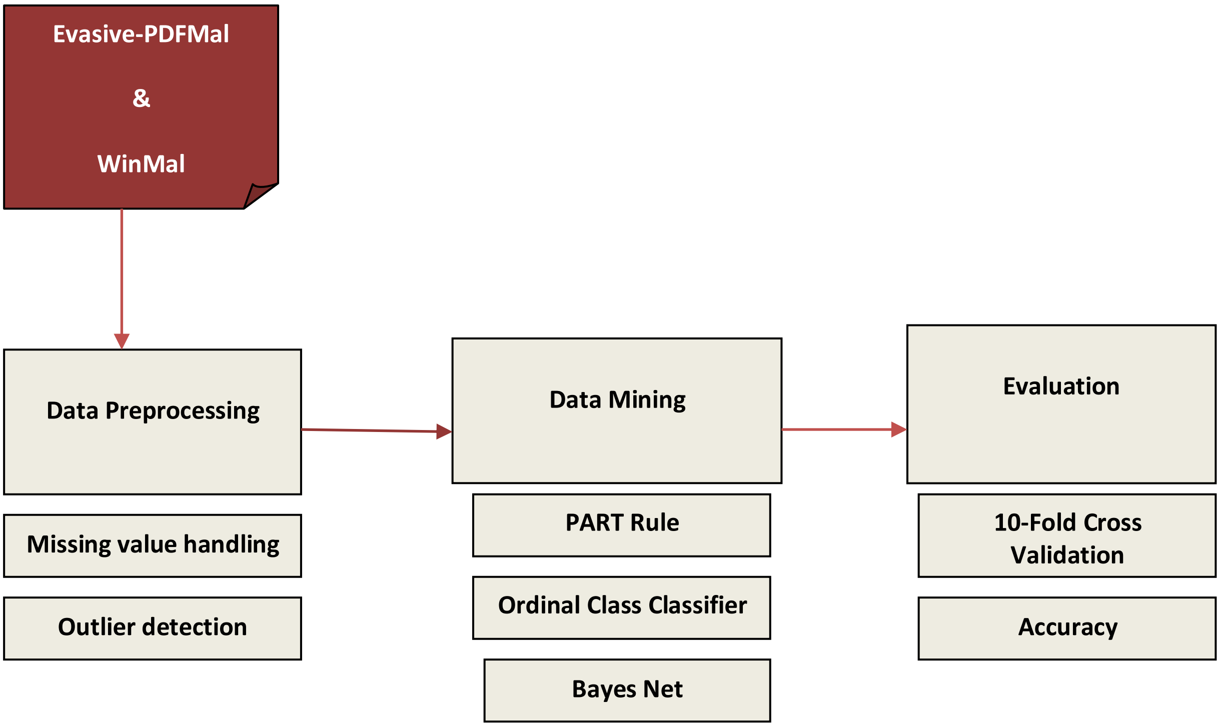

Figure 1 – Design specification

5 Implementation

In this research, Waikato Environment for Knowledge Analysis (WEKA) was employed. It is an open-source classifier programme. It is a popular machine learning programme for mining and data analysis. The Java script software was developed by the University of Waikato in New Zealand. WEKA's ability to handle several data formats, including JavaScript Object

Notation (JSON) and comma-separated values (CSV), is one of its advantages. Attribute Relation File Format (ARFF) is still the default file format.

The analysis and implementation of machine learning algorithms were conducted using the WEKA tool. WEKA, a widely acknowledged data mining and machine learning software, facilitated the experimentation with various algorithms and datasets, offering a user-friendly environment for analysis and assessment.

6 Evaluation

6.1 Results and Discussion

Table VI, VII, and VIII shows the results of the four datasets that were used and the machine learning algorithms. This section presents the results of the models in terms of the accuracy, precision and recall.

A. Model Performance

The findings of the three different machine learning algorithms are displayed in table VI with ten k-fold cross-validations. Figure 3 compares the performance of PART, BN and OCC model in terms of accuracy, precision and recall. The result shows that PART and OCC had similar performance.

In terms of absolute values, the OCC and PART models have similar accuracy values but the OCC model has a slightly higher accuracy for the four datasets. The OCC model and the PART model have the highest accuracy of (100%) for the DDLs_Imported dataset. Accuracy percentage is not the only factor that determines high scores; other factors include recall and precision.

Table VI – Performance results for PART models using 10 fold Cross Validation

Dataset	Accuracy(%)	Precision	Recall	F1
Evasive-PDFMal	98.95	0.991	0.989	0.990
WinMal (PE_Header)	99.27	0.676	0.885	0.767
WinMal (PE_Section)	99.98	1.000	1.000	1.000
WinMal (DLLs_Imported)	100	1.000	1.000	1.000

Table VI1 – Performance results for OCC models using 10 fold Cross Validation

Dataset	Accuracy(%)	Precision	Recall	F1
Evasive-PDFMal	99.00	0.989	0.992	0.991
WinMal (PE_Header)	99.23	0.992	1.000	0.996
WinMal (PE_Section)	99.82	0.999	1.000	1.000
WinMal (DLLs_Imported)	100	1.000	1.000	1.000

Table VIII – Performance results for BN models using 10 fold Cross Validation

Dataset	Accuracy(%)	Precision	Recall	F1
Evasive-PDFMal	97.17	0.977	0.970	0.974
WinMal (PE_Header)	97.63	0.996	0.980	0.988

WinMal (PE_Section)	98.33	0.999	0.984	0.991
WinMal (DLLs_Imported)	99.98	1.000	1.000	1.000

6.2 Comparison with Previous Research

The performance of the ML models were compared with previous papers (Issakhani et al., 2022) (Yousuf et al., 2023) to know we if can get similar results on the datasets. The same validation process and the data split percentage from the previous studies was used for the datasets. Table IX and X has the summary of the results. For the Evasive-PDFMal dataset, PART performed better with an accuracy of 98.96% and 98.95% in a 5 fold and 10 fold validation data split. For the WinMal dataset, PART also achieved a slightly better accuracy of 99.98 in 70:30% data split.

TABLE IX – Comparison with results from previous studies on Evasive-PDFMal dataset

Reference	**Dataset**	**Data Split**	**ML Algorithm**	**Accuracy(%)**
(Issakhani et al., 2022)	Evasive-PDFMal	5f-CV	AdaBoost	97.32
Our Results	Evasive-PDFMal	5f-CV	PART	**98.96**
(Yerima & Bashar, 2023)	Evasive-PDFMal	10f-CV	AdaBoost	98.83
Our Results	Evasive PDFMal	10f-CV	PART	**98.95**

TABLE X – Comparison with results from previous studies on WinMal dataset

Reference	**Dataset**	**Data Split**	**ML Algorithm**	**Accuracy(%)**
(Yousuf et al., 2023)	WinMal (DLLs_Imported)	70:30%	Random Forest	96.41
Our Results	WinMal (DLLs_Imported)	70:30%	PART	**99.98**
(Yousuf et al., 2023)	PE_Header	70:30%	Random Forest	99.36
Our Results	PE_Header	70:30%	PART	**99.32**
(Yousuf et al., 2023)	PE_Section	70:30%	Random Forest	97.32
Our Results	PE_Section	70:30%	PART	**99.98**

6.3 Model Building Time

The model building time are shown in seconds per 1000 samples on the WEKA tool. Figure 2 presents the model building time for both datasets and their ML algorithms. The figure shows that the model bulding time increased as the size of the dataset increased. On the evasive-PDFMal, OCC has a lower building time of 0.17s than the model time of 2.18s in the WinMal (PE_Section) dataset. This could be as a result of the size of the dataset. The larger the size of the dataset, the more time it will take to build a model. It can also be said that the model time decreases significantly as the number of features decreases.

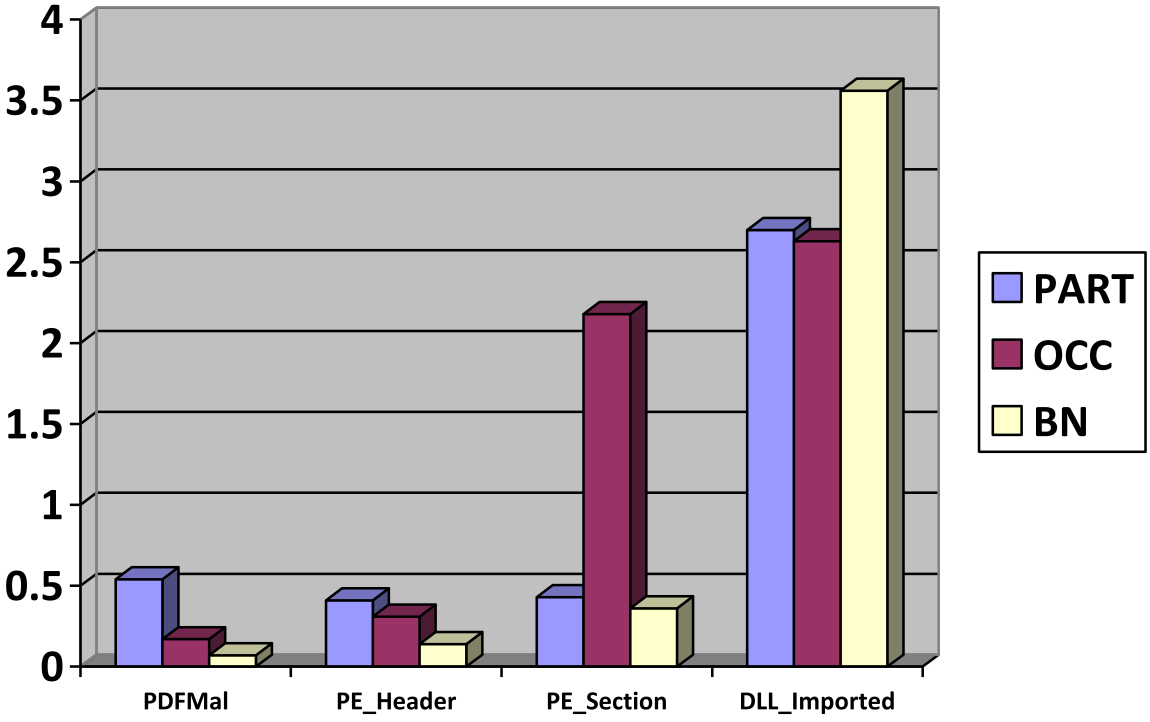

Fig 2 – Time taken to build models

6.4 Discussion

This research made use of multiple classifiers to train and test the malware datasets. Ten k-fold cross-validations were used to evaluate and train the machine learning algorithm. This allowed it to predict for every instance in the dataset and produce a result that is the sum of all of the individual predictions. In order to achieve the required accuracy, it was repeated on the classifiers. Three models were built in order to have the highest level of precision and accuracy. Model performance and predictive ability must be understood in order to assess the models using metrics like accuracy, precision, and recall.

This study made use of the primary evaluation report which includes the model's recall, accuracy, and precision for every malware dataset. The outcomes of the malware detection experiments were good, with accuracy scores ranging between 97% and 100%. The model with the highest accuracy on the Evasive-PDFMal dataset was the Ordinal Class Classifier (OCC), the model achieved an accuracy of 99% on the dataset. The model with the highest accuracy on the WinMal dataset was the Ordinal Class Classifier (OCC) and PART Rule (PART). Both models achieved 100% accuracy on the WinMal (DLLs_Imported) dataset.

These results showed the efficacy of the chosen machine learning algorithms in accurately detecting malware within PE and PDF files. The use of machine learning algorithms in accurately detecting malware in PE and PDF files is very effective. This solution is effective because I made use of different datasets which includes the Evasive PDF Mal and the WinMal files (PE_Header, PE_Section and DLLs_Imported). The combination of the datasets resulted in a large dataset to work with and it represents the data of PE and PDF files. The research question and objectives of this project has been answered and achieved.

Additionally, the methodology adopted a comprehensive approach, combining different datasets, using a range of machine learning algorithms, and employing the WEKA tool for analysis. This methodology provided valuable insights into the efficiency of the selected algorithms and also offered a foundation for future research in the field of malware detection.

7 Conclusion and Future Work

The purpose of this research is to evaluate how accurately supervised machine learning algorithms can detect malware in PE files and PDF files. Four datasets were selected and analysed on the WEKA tool. This approach was efficient and these datasets contained a diverse range of Windows malware samples and PDF files, offering a good testing ground for the effectiveness of supervised machine learning algorithms. This research used three machine algorithms, PART, Bayes Net, and OCC.

According to the result, OCC and PART model achieved a detection accuracy of 100% on one of the WinMal dataset while BN achieved an accuracy of 99.98%. This means that PART and OCC has greater accuracy rate when scanning harmful PE files and PDF files. Compared to other research papers that used the same datasets that was used in this study, this study achieved the highest level of accuracy. This research has demonstrated the ability of the machine learning algorithms to accurately detect malware in PE and PDF files.

Through further research and future work, real-time protection capabilities should be built to prevent malware threats. Due to the fact that malware evolves constantly and some new changes will be observed, there will be need to retrain the trained models after a while. The models can be trained through periodic update, continuous monitoring and learning of new datasets so that the trained models can be up to date with the evolving nature of malware. The machine learning models will also need to be trained on a larger dataset and good performing models should be able to detect the new variants of malware. The idea of concept drift and dynamic machine learning algorithms with drifting abilities for real-time protection against malware should be explored.

References

Akhtar, M. S., & Feng, T. (2023). Evaluation of Machine Learning Algorithms for Malware Detection. *Sensors (Basel, Switzerland)*, *23*(2), 946. https://doi.org/10.3390/s23020946

Almomani, I., AlKhayer, A., & Ahmed, M. (2021). An Efficient Machine Learning-based Approach for Android v.11 Ransomware Detection. *2021 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA)*, 240–244. https://doi.org/10.1109/CAIDA51941.2021.9425059

Al-Taani, R., Bassah, R., Naimat, N., & Odeh, A. (2023). PDF Malware Detection optimisation using machine learning. *2023 3rd International Conference on Computing and Information Technology (ICCIT)*, 15–19. https://doi.org/10.1109/ICCIT58132.2023.10273942

Anderson, H., & Roth, P. (2018). EMBER: An Open Dataset for Training Static PE Malware Machine Learning Models. *ArXiv*. https://www.semanticscholar.org/paper/7e152e587fbc73b5bd23048c71c1b36c569416c5

Asaju, C. B., Otoo-Arthur, D., Orah, R. O., & Sekyi-Dadson, F. (2021). Development of a Machine Learning Model for Detecting and Classifying Ransomware. *2021 1st International Conference on Multidisciplinary Engineering and Applied Science (ICMEAS)*, 1–5. https://doi.org/10.1109/ICMEAS52683.2021.9692402

Aslan, Ö., & Yilmaz, A. A. (2021). A New Malware Classification Framework Based on Deep Learning Algorithms. *IEEE Access*, *9*, 87936–87951. https://doi.org/10.1109/ACCESS.2021.3089586

Barut, O., Grohotolski, M., DiLeo, C., Luo, Y., Li, P., & Zhang, T. (2020). Machine Learning Based Malware Detection on Encrypted Traffic: A Comprehensive Performance

Study. *Proceedings of the 7th International Conference on Networking, Systems and Security*, 45–55. https://doi.org/10.1145/3428363.3428365

Barut, O., Zhang, T., Luo, Y., & Li, P. (2023). A Comprehensive Study on Efficient and Accurate Machine Learning-Based Malicious PE Detection. *2023 IEEE 20th Consumer Communications & Networking Conference (CCNC)*, 632–635. https://doi.org/10.1109/CCNC51644.2023.10060214

Faruk, M. J. H., Masum, M., Shahriar, H., Qian, K., & Lo, D. (2022). Authentic Learning of Machine Learning to Ransomware Detection and Prevention. *2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC)*, 442–443. https://ieeexplore.ieee.org/abstract/document/9842537/

Feyyad, U. M. (1996). Data mining and knowledge discovery: Making sense out of data. *IEEE Expert*, *11*(5), 20–25. https://doi.org/10.1109/64.539013

Gusti, A., & Girinoto, G. (2023). *PDFalse: Evasive Malicious PDF Machine Learning Classifier*. 9–14. https://doi.org/10.1109/ICoCICs58778.2023.10277336

He, D. (2022). Active learning for ordinal classification based on expected cost minimization. *Scientific Reports*, *12*(1), Article 1. https://doi.org/10.1038/s41598-022-26844-1

Herrera-Silva, J., & Alvarez, M. (2023). Dynamic Feature Dataset for Ransomware Detection Using Machine Learning Algorithms. *Sensors*, *23*, 1053. https://doi.org/10.3390/s23031053

Issakhani, M., Victor, P., Tekeoglu, A., & Lashkari, A. (2022). PDF Malware Detection based on Stacking Learning: *Proceedings of the 8th International Conference on Information Systems Security and Privacy*, 562–570. https://doi.org/10.5220/0010908400003120

Jeyalakshmi, V. S., Jayapriya, J., & Krishnan, N. (2022). A Study of Malware Datasets and Techniques to Detect the Malware using Deep Learning Approach. *2022 6th*

International Conference on Trends in Electronics and Informatics (ICOEI), 856–862. https://doi.org/10.1109/ICOEI53556.2022.9777220

Kaur, J., & Ramkumar, K. . R. (2022). The recent trends in cyber security: A review. *Journal of King Saud University - Computer and Information Sciences*, *34*(8, Part B), 5766–5781. https://doi.org/10.1016/j.jksuci.2021.01.018

Kumar, A., Kuppusamy, K. S., & Aghila, G. (2019). A learning model to detect maliciousness of portable executable using integrated feature set. *Journal of King Saud University - Computer and Information Sciences*, *31*(2), 252–265. https://doi.org/10.1016/j.jksuci.2017.01.003

Li, Y., & Liu, Q. (2021). A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments. *Energy Reports*, *7*, 8176–8186. https://doi.org/10.1016/j.egyr.2021.08.126

Martina Jose Mary, S., U., P., M. B., & Sandhya, S. G. (2020). Detection of ransomware in static analysis by using Gradient Tree Boosting Algorithm. *2020 International Conference on System, Computation, Automation and Networking (ICSCAN)*, 1–5. https://doi.org/10.1109/ICSCAN49426.2020.9262315

Masum, M., Hossain Faruk, M. J., Shahriar, H., Qian, K., Lo, D., & Adnan, M. I. (2022). Ransomware Classification and Detection With Machine Learning Algorithms. *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, 0316–0322. https://doi.org/10.1109/CCWC54503.2022.9720869

Poudyal, S., Subedi, K. P., & Dasgupta, D. (2018). A Framework for Analyzing Ransomware using Machine Learning. *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, 1692–1699. https://doi.org/10.1109/SSCI.2018.8628743

Prajapati, P., & Stamp, M. (2021). *An Empirical Analysis of Image-Based Learning Techniques for Malware Classification*. 411–435. https://doi.org/10.1007/978-3-030-62582-5_16

Ramadhan, F. H., Suryani, V., & Mandala, S. (2021). Analysis Study of Malware Classification Portable Executable Using Hybrid Machine Learning. *2021 International Conference on Intelligent Cybernetics Technology & Applications (ICICyTA)*, 86–91. https://doi.org/10.1109/ICICyTA53712.2021.9689130

Shi, Y., Li, P., Yuan, H., Miao, J., & Niu, L. (2019). Fast kernel extreme learning machine for ordinal regression. *Knowledge-Based Systems*, *177*, 44–54. https://doi.org/10.1016/j.knosys.2019.04.003

Shroff, K. P., & Maheta, H. H. (2015). A comparative study of various feature selection techniques in high-dimensional data set to improve classification accuracy. *2015 International Conference on Computer Communication and Informatics (ICCCI)*, 1–6. https://doi.org/10.1109/ICCCI.2015.7218098

Smith, D., Khorsandroo, S., & Roy, K. (2023). Supervised and Unsupervised Learning Techniques Utilizing Malware Datasets. *2023 IEEE 2nd International Conference on AI in Cybersecurity (ICAIC)*, 1–7. https://doi.org/10.1109/ICAIC57335.2023.10044169

Tang, Y., Dong, J., Guo, Y., Zhou, Y., Lu, F., & Zhang, B. (2022). A Malicious PDF File Detection Method Based on Improved Ensemble Learning Stacking. *2022 4th International Conference on Frontiers Technology of Information and Computer (ICFTIC)*, 475–478. https://doi.org/10.1109/ICFTIC57696.2022.10075332

Trad, F., Hussein, A., & Chehab, A. (2023). Leveraging Adversarial Samples for Enhanced Classification of Malicious and Evasive PDF Files. *Applied Sciences*, *13*(6), Article 6. https://doi.org/10.3390/app13063472

Ucci, D., Aniello, L., & Baldoni, R. (2019). Survey of machine learning techniques for malware analysis. *Computers & Security*, *81*, 123–147. https://doi.org/10.1016/j.cose.2018.11.001

Upchurch, J., & Zhou, X. (2015). Variant: A malware similarity testing framework. *2015 10th International Conference on Malicious and Unwanted Software (MALWARE)*, 31–39. https://doi.org/10.1109/MALWARE.2015.7413682

Yerima, S. Y., & Bashar, A. (2023). Explainable Ensemble Learning Based Detection of Evasive Malicious PDF Documents. *Electronics*, *12*(14), Article 14. https://doi.org/10.3390/electronics12143148

Yousuf, M. I., Anwer, I., Riasat, A., Zia, K. T., & Kim, S. (2023). Windows malware detection based on static analysis with multiple features. *PeerJ Computer Science*, *9*, e1319. https://doi.org/10.7717/peerj-cs.1319