Title
Predictive Modelling for Early Detection and Prevention of Ransomware, and Malware Using Machine Learning

MSc Research Project
Programme Name:
Srilakshmi Thota

Forename Surname
Thota Srilakshmi
Student ID:
22194916

School of Computing
National College of Ireland

Supervisor: Jawad Salahuddin

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Thota Srilakshmi		
Student ID:	22194916		
Programme:	MSc Cyber-Security	**Year:**	2024
Module:	Thesis		
Supervisor:	Jawad Salahuddin		
Submission Due Date:	12/08/2024		
Project Title:	Predictive Modelling for Early Detection and Prevention of Ransomware, and Malware Using Machine Learning		

Word Count: 10543 **Page Count**:22

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: T Srilakshmi

Date:12/08/2024

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):	

Predictive Modelling for Early Detection and Prevention of Ransomware, and Malware Using Machine Learning

Abstract

In the digital arena, the growing frequency of ransomware and malware attacks makes efficient detection and mitigating techniques ever more crucial. This study focuses on machine learning techniques for ransomware and virus detection. We want to develop detection models that, with the help of advanced algorithms and preprocessing methods, can correctly identify dangerous software. The Random Forest model outperformed all the other models with high accuracy and F1-score. Other methods like K-Nearest Neighbors (KNN) and Support Vector Machine (SVM) also performed very well the accuracy of KNN was close to one while, using methods such as SMOTE and ADASYN, SVM also exhibited high level of accuracy. Other techniques, such as Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) provided much higher accuracy, 99. As for instance, SMOTE achieved an average accuracy of 98% confirming its capacity for handling data despite having been synthesized for sequential pattern data. Logistic Regression was the most accurate with a percentage of 93.83%. These findings demonstrate the effectiveness of sophisticated machine learning models in the detection of malware and ransomware. The solution that we have made is a response system. When this system is deployed into any kind of environment it will help in monitoring the system. In that the API can be integrated to any enterprise server. This response system monitors in such a way that the solution can generate the log files of the intrusions or suspicious activity in th form of malware or ransomware.

Keywords: *Machine learning, Ransomware and malware, Logistic, SVM, Cybersecurity, Modeling*

Chapter 1: Introduction

The spread of several kinds of virus, including ransomware, seriously affects people and businesses both now in the digital terrain. Particularly ransomware has become well-known as among the most obvious and powerful types of malware. This harmful programme locks access to important data by encrypting files on a victim's computer system and demands a ransom for the decryption key. The growing frequency of ransomware attacks emphasises how urgently strong detection and prevention systems are needed to protect private data and preserve operational integrity (**Conti, M., Dehghantanha, A., Franke, K. and Watson, S., 2018**).

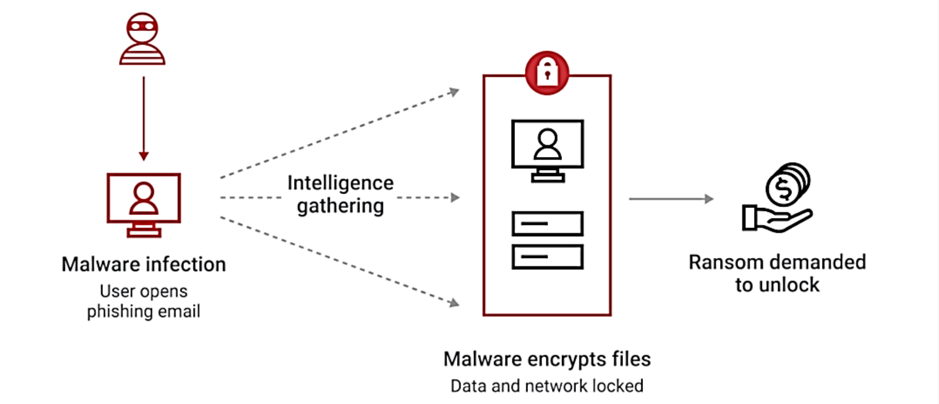
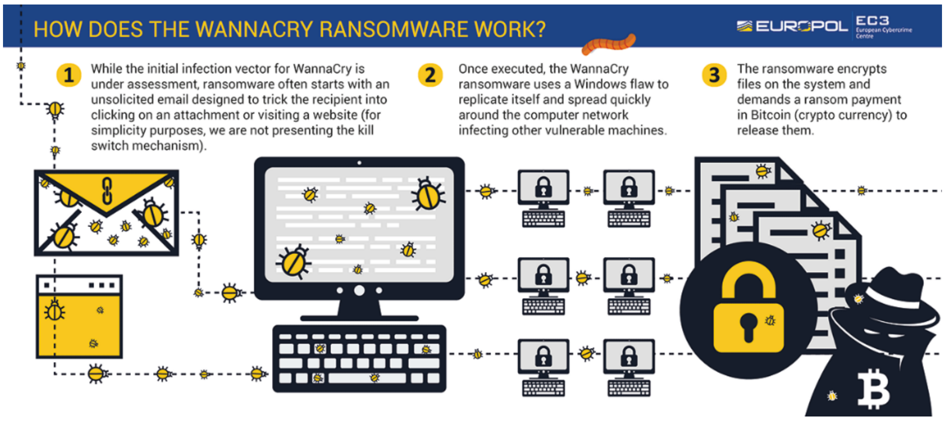


Figure 1: Ransomware and Malware (**Source: Akamai**)

Rising in number and complexity, ransomware assaults frequently target weak systems with devastating efficiency. These strikes can destroy systems of local and state governments, disturb vital services, and cause significant financial losses to impacted companies. The infamous 2017 "Wannacry" ransomware campaign reminds us sharply of the possible scope and profitability of such cyber extortion operations (**Kouliaridis, V., Barmpatsalou, K., Kambourakis, G. and Chen, S., 2020**). This well reported incident made clear how vulnerable digital infrastructure is and how urgently strong cybersecurity is needed. The ransomware threat scene was further worsened by the COVID-19 epidemic since remote work sit- uations became the standard and cybersecurity defences were strained (**Muniandy, M., Ismail, N., Al-Nahari, A. and Ngo Yao, D., 2024**). ransomware attacks surged dramatically over the period; a 50% increase in the latter half of 2020 compared to the first half of the year is recorded (**Humayun, M., Jhanjhi, N.Z., Alsayat, A. and Ponnusamy, V., 2021**). Remote work arrangements and the use of pandemic-related vulnerabilities by cyber- criminals help to explain this jump in attack surface provided. Early ransomware incidents at- tracted a lot of interest around 2013, and since then, these assaults have seriously disrupted operations and finances (**Kharraz, A., et. al., 2023**). Sophisticated capabilities of modern ransomware that take advantage of security weak- nesses make identification and prevention very difficult (**Chesti, I.A., et. al., 2020**).


HOW DOES THE WANNACRY RANSOMWARE WORK?
EUROPOL
EC3
European Cybercrime Centre
1
2
3
B

While the initial infection vector for WannaCry is under assessment, ransomware often starts with an unsolicited email designed to trick the recipient into clicking on an attachment or visiting a website (for simplicity purposes, we are not presenting the kill switch mechanism).

Once executed, the WannaCry ransomware uses a Windows flaw to replicate itself and spread quickly around the computer network infecting other vulnerable machines.

The ransomware encrypts files on the system and demands a ransom payment in Bitcoin (crypto currency) to release them.

Figure 2: WannaCry 2017 Ransomware working (**Source: Europol**)

The application of machine learning methods for the ransomware and malware detection is investigated in this paper. Data preparation and oversampling call for Synthetic Minority Over-sampling Technique (SMote), Adaptive Synthetic Sampling (ADASYN), and K-Means clustering. These methods guarantee that our models are trained on balanced data and can thus efficiently identify both common and rare threats, hence addressing the problem of class imbalance that is widespread in cybersecurity datasets.

1.1 Research Problem

Because both harmful software and cybercriminals' strategies are always evolving, cybersecurity researchers face a significant problem when trying to detect ransomware and other forms of malware. The sophistication and velocity of emerging malware variants makes it difficult for traditional detection methods relying on rule-based systems and signature matching to stay up. Therefore, more efficient and adaptive detection methods that reliably and accurately identify harmful software are urgently required.

1.2 Motivation

This study is motivated by the important need of reducing the hazards presented by malware and ransomware to people, companies, and society in general. The possible influence of cyberattacks has become much more pronounced as digital technologies proliferate and systems' interconnection rises. We want to provide proactive and intelligent systems that can detect malware and ransomware in real-time by using machine learning approaches, therefore allowing quick response and mitigating action to guard against cyber attacks (**Sood, A.K. and Enbody, R.J., 2013**).

1.3 Background

Financial systems are now more interconnected due to the integration of world economies and information technology breakthroughs, but they are also more vulnerable to sophisticated fraud schemes (**Choudhary, S. and Sharma, A., 2020**). Conventional monitoring methods are inadequate for identifying novel and developing fraud trends since they frequently depend on static rules and previous data. Earlier studies have looked into a number of strategies to improve fraud detection, such as rule-based systems, machine learning models, and statistical analysis. However, because financial fraud is dynamic, detection techniques must constantly innovate (**Suarez-Tangil, G., Tapiador, J., Peris-Lopez, P. and Ribagorda, A., 2013**). By investigating the application of cutting-edge machine learning approaches, such as hybrid resampling methods and clustering models, to increase fraud detection accuracy, this work expands on prior studies.

1.4 Research Solution

Our work suggests the identification of malware and ransomware using advanced preprocessing methods and machine learning algorithms (**Bae, S., Lee, G. and Im, E.G., 2019**). We want to build strong detection models able to precisely identify harmful software based on typical patterns and behaviours by using supervised and unsupervised learning approaches. We also investigate the class imbalance inherent in malware datasets by means of oversam- pling methods such Synthetic Minority Oversampling Technique (SMote) and Adaptive Synthetic Sam- pling (ADASYN). Moreover, we look at the possible advantages of using clustering techniques including K-Means to improve latent structural capture in the data and feature representation.

1.5 Research Question

RQ1: How can machine learning methods be efficiently applied to detect ransomware and malware?

RQ2: What is the impact of incorporating clustering techniques on feature representation and model performance in malware and ransomware detection settings?

Chapter 2: Literature Review

2.1 Ransomware Detection Techniques

The thorough analysis by Amjad Alraizza (**Alraizza, A. and Algarni, A., 2023**) discusses the detrimental effects of ransomware attacks on computer systems and private information. These attacks frequently lead to data destruction, disclosure, and unauthorised access, which can cause significant financial losses as well as harm to one's reputation. Alraizza highlights the importance of precise, timely, and dependable detection techniques. The survey highlights the growth and growing sophistication of these threats by providing a historical context and history of ransomware assaults. It identifies prospective study topics and unresolved difficulties that want more investigation while highlighting the most recent developments in automated ransomware detection, prevention, mitigation, and recovery. This poll is an essential tool for learning about the state of auto- mated ransomware detection today and provides information on how to lessen the effects of ransomware on people and companies. Using machine learning and deep learning techniques, A. Charmilisri and Ineni Harshi (**Charmilisri, A., Harshi, I., Madhushalini, V. and Raja, L., 2023**) present a novel method for detecting ransomware viruses. In order to gain a better understanding of ransomware attacks, the study looks into the behavioural traits of both Normal and hazardous applications. A machine learning model's random dataset classifier is used to train the data. The method helps users scan files and spot ransomware threats by using ensemble detection techniques and random datasets. Users are better able to identify dangers and take preventative action when files are consistently scanned. The authors propose utilising larger datasets and adding more Normalapplications to investigate different widely used apps, improving detection accuracy and robustness, in order to reduce overfitting. This work emphasises the value of behavioural analysis in ransomware detection and makes recommendations for future research to improve the robustness and accuracy of detection. Given the dearth of real-world ransomware samples, Md Shazzadur Rahman, Md Sayeed Ahmed Sab- bir, and Sudipto Ghosh (**Rahman, M.S., Sabbir, M.S.A. and Ghosh, S., 2024**) emphasise the significance of data preparation and augmentation tactics to enhance the generalisation power of machine learning models. Effective methods for handling imbalanced datasets were studied, including data augmentation, under- and over-sampling, and over-sampling. Sev- eral classifiers were used in the study, and Random Forest had the best accuracy (99.7%), closely followed by Decision Tree (99.5%)..

Using supervised machine learning, Yotam Mkandawirea and Aaron Zimba (**Mkandawire, Y. and Zimba, A., 2023**) present a reliable method for identifying malware in Windows executable files. Their methodology successfully detects ransomware at the host level by fusing dynamic malware analysis with supervised learning. The study used a vari- ety of machine learning algorithms and feature extraction techniques, with Random Forest and Gradient Boosting demonstrating the greatest promise. The suggested system proved to be highly dependable and efficient, indicating that real-time application of it is possible. The framework will be integrated with the current cybersecurity infrastructure in the future, and new features will be investigated to increase detec- tion precision and lower false positives. The need to create systems that can accurately identify both known and novel types of ransomware is imperative given the growing threat posed by cyberattacks. In order to get optimal results, Majd and Mazumdar (**Majd, N.E.M. and Mazumdar, T., 2023**) included different feature selection techniques, machine learning (ML) and deep learning (DL) algorithms, along with hyperparameter tweaking, in their models and extensive tests.

The results of this investigation (**Wadho, S.A., Yichiet, A., Gan, M.L., Lee, C.K., Ali, S. and Akbar, R., 2024**) highlight how crucial machine learning is to improving ransomware detection systems. After a thorough evaluation of several methods, Support Vector Machines (SVM) was shown to be the most effective classifier with a high accuracy rate. The findings highlight how crucial fea- ture engineering is for improving the model's discriminative abilities, especially when it comes to API call attributes. This study uses machine learning to further the continuing attempts to strengthen cybersecurity defences. Through concentrating on enhancing the precision and versatility of ransomware identification systems, the study seeks to deliver concrete advantages in the shape of more resilient and anticipatory cybersecurity remedies.

2.2 Malware Detection and Categorization

A lightweight, efficient, and precise machine-learning-based method for malware detection and classification is presented by Attaullah Buriro as MalwDC (**Buriro, A., Buriro, A.B., Ahmad, T., Buriro, S. and Ullah, S., 2023**). Using the BODMAS dataset, the method was tested in two scenarios: malware classification (multi-class classification) and malware detection (Normalvs. malware). With an accuracy of 99, the random forest classifier proved to be the most efficient. In the malware detection scenario, 38% of all features and 99.56% of a subset of 25 features were de- tected. The method achieved 97.59% accuracy on full features and 97.69% accuracy on chosen features for malware categorization. The suggested approach showed promise for practical uses with its high ac- curacy and quick training and testing periods.

Amit Kumar Bairwa, Priyanshu Kumar, Akshit Kamboj, and others investigated supervised and unsuper- vised machine learning models for malware detection using Principal Component Analysis (**Kamboj, A., Kumar, P., Bairwa, A.K. and Joshi, S., 2023**), Random Forest, Decision Tree, Logistic Regression, and K-Means clustering, among other methods. The study used SMOTE, balanced bagging classifier, oversampling, undersampling, and other balancing dataset strategies to handle unbalanced data concerns. With an accuracy of 99.99%, the Random Forest model was the most accurate. When compared to earlier studies, the balanced dataset procedures performed better, under- scoring the significance of data handling strategies in enhancing model accuracy and reliability. Using sophisticated machine learning models, such Convolutional Neural Networks (CNNs) and graph- based representations, to identify intricate patterns in malware data is becoming more and more popular. Using these models in realistic, real-world cybersecurity scenarios requires large-scale dataset evaluations and real-time detection capabilities. By combining these cutting-edge meth- ods, malware detection is expected to advance, leading to faster and more precise threat identification. Mazin Abed Mohammed (**Mohammed, M.A., 2019**) examined how hyper-parameters and machine learning could be used to increase the security of medical data. In order to improve accuracy for heartbeat data in the Ifogsim simulation environment, the study created a lightweight distributed training and testing level Adaptive Machine Learning (AMDML) model to handle training and testing delays. When compared to centralised machine learning models, the AMDML model performed better, indicating that it has practical benefits in healthcare data security.

In order to detect ransomware, Karam Hammadeh and Kavitha M (**Hammadeh, K. and Kavitha, M., 2023**) investigated sophisticated machine learning methods like Support Vector Machine (SVM), Logistic Regression, K-Nearest Neighbours (KNN), and Long Short-Term Memory (LSTM). With an accuracy of 99.08%, the LSTM model was most success- ful in recognising complex patterns and temporal dependencies in the data. In order to increase accuracy and resilience, the study suggests expanding the dataset to include features from both static and dynamic analysis, as well as investigating hybrid models like CNN-LSTM. These advanced algorithms and composite models can help to enhance the method of ransomware identification and decrease the consequences of ransomware attacks. From the literature study, we can see that machine, learning plays a significant role in enhancing the identification of malware and ransomware. Although, enhanced models like the CNNs, and LSTMs exhibited good potentials in the nex- t future improvements, some normal models such as the Random Forest and the SVM have proven to work accurately and quite efficiently.

2.3 Research Summary

Table 1: Summary of Papers

Sl. No	Paper Name	Authors Name	Dataset Used	Models Used	Results Sum- mary
1	Ransomware Detection Using Machine Learn- ing: A Survey	Amjad Alraizza	-	Random Forest, SVM, KNN, XGBoost, Logis- tic Regression, Decision trees	-
2	A Quick and Ac- curate Machine Learning-Based Approach for Malware De- tection and Categorization	Attaullah Buriro	BODMAS	MalwD&C	It achieved an accuracy of 99.38% on full features and 99.56% on a selected subset of 25 features
3	A Novel Ran- somware Virus Detection Tech- nique using Machine and Deep Learning Methods	A. Charmilisri, Ineni Harshi	Microsoft App Store, VirusShare.com	Deep ensemble model	The deep en- semble model was able to achieve an ac- curacy of over 80% for the alternative hy- pothesis when categorising PE files as ran- somware or not

4	Detection of malware in downloaded files using var-ious machine learning models	Akshit Kamboj, Priyanshu Kumar,Amit Kumar Bairwa	Different datasets merged	Random Forest, Decision Tree, SVM, K-Means, Logistic regression, one hot encoding	Random Forest Model wasfound to be the most accurate. Its accuracy went as high as 99.99% for the test dataset
5	Ransomware Attack Detection using Machine Learning Approaches	Md Shazzadur Rahman; Md Sayeed Ahmed Sabbir; Sudipto Ghosh	CICAndMal 2017	SVM, XGB, Random Forest, and Decision Tree	Model is incredibly precise and has outstanding performance accuracy since itis built on SVM and uses a Gaussian kernel.
6	Ransomware Detection Techniques Using Machine Learning Methods	Shuaib Ahmed Wadho; Aun Yichiet	Kaggle.com	SVM, KNN, Naive Bayes, Linear Model, Decision Tree, and Random Forest.	The meticulous assessment of various algorithms uncovered that SVM arose as the most successful classifier,
7	A Supervised Machine Learning Ransomware Host- Based DetectionFramework	Yotam Mkandawireaand, Aaron Zimba	-	Decision Tree Classifier, Random Forest Classifier, Gradient Boosting Classifier, Ada Boost Classifier, Gaussian Naïve Bayes, and Logistic Regression	Logistic Regression algorithm model witha 97.7% accuracy score offers a 99% success rate in ransomware detection
8	Adaptive secure malwareefficient ma-chine learning algorithm for healthcare data	Mazin Abed Mohammed	Github, Kaggle	Random forests, CNN, SV M,AMDML	AMDML outperforms machine learning mal- ware analysis models in terms of accuracy by60%, delay by 50%, and detection of original heartbeat data by 66%
9	Unraveling Ransomware: De- tecting Threats with Advanced Machine Learning Algorithms	Karam Hammadeh; Kavitha,M	-	LSTM, SVM, LR, KNN	LSTM model, has shown remarkable accuracy, reaching up to 99%, in detecting malware and ransomware
10	Ransomware Classification Using Machine Learning	Nahid Ebrahimi Majd; Torsha Mazumdar	Github	DT, RF, LR, NB, SVM, KNN, XGB, MLP and CNN	RF and MLP with Filter method and SVM with Wrap- per method

					were the best 3 models in terms of accuracy, precision, recall, and f1-score

2.4 Research Niche

As we look into the above results, we find hat machine learning is being in use and the results are generated. But here we focus on a pipeline. This pipeline

a. Maintainns and train from a bag of models generated the best model based on the greedy selection
b. This response system can be integrated to any system, environment and server to monitor continuous and generate the log files

This kind of response system is not discussed anywhere and we are focusing in developing that kind of response system.

Chapter 3: Methodology

3.1 Handling Class Imbalance

SMOTE: A big part of the Synthetic Minority Over-sampling Technique (SMOTE) (**Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P., 2002**) is how it fixes the class mismatch problem in datasets when machine learning is used to find ransomware and other malware. This case has a skewed class distribution because the minority class is mostly made up of examples of bad behaviour. SMOTE fixes this problem by making fake examples for the minority class.

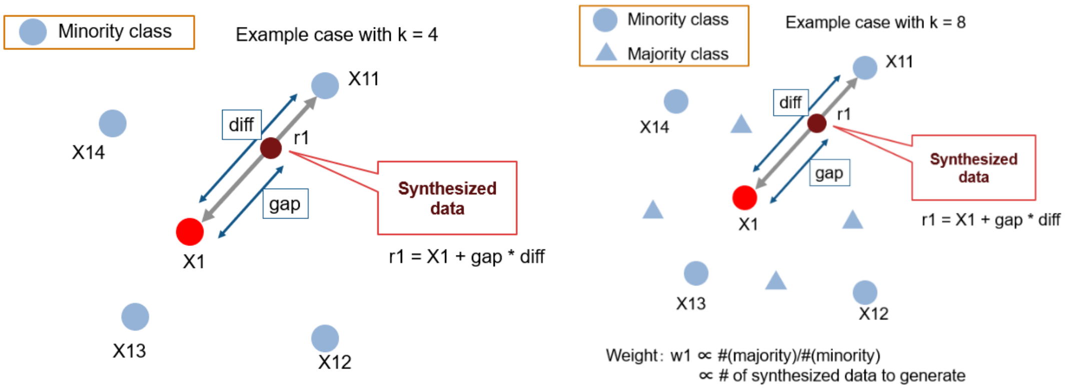


Figure 3: Working of SMOTE (left) and ADASYN (right) comparison. Introduction of synthetic data into the correct data

ADASYN: A further effective strategy for managing class imbalance in datasets used for ransomware and malware detection is the Adaptive Synthetic (ADASYN) sampling method (**He, H., Bai, Y., Garcia, E. and Li, S., 2008**). Based onthe density distribution, ADASYN creates synthetic data points for the minority class (suspicious case behaviours).

In contrast to SMOTE, ADASYN focuses more on producing data for more difficult-to-learn minority class examples, resulting in a more evenly distributed set of synthetic samples. This method enhances the classifier's capacity to correctly detect and classify malware and ransomware and helps the model learn from challenging cases. The training dataset is more balanced when ADASYN is used, which improves the model's robustness and dependability in identifying harmful activity.

Stratification: In the case of the training and the assessment set, one-half of the transactions should be suspicious (malware or ransomware) while the other half should not be suspicious, and this is true if stratification is used. This technique entails arranging the dataset according to the class labels so that there is a balance of both the suspicious (malware or ransomware) and non- events when training the model so that the model does not over-rely on the majority class. This balance enhances the model's ability to learn discriminative features for fraud while at the same time possessing high classification throughput rate.

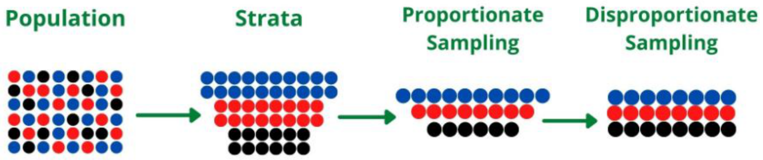


Figure 4: Stratification of the dataset (**He, H., Bai, Y., Garcia, E. and Li, S., 2008**)

3.2 Machine Learning Algorithms

__**Random Forest**__: Strong ensemble learning techniques like Random Forest (**Ali, J., Khan, R., Ahmad, N. and Maqsood, I., 2012**) build several decision trees during training and aggregate their predictions by generating the class mode for classification tasks. Ran- dom Forest models are widely recognised for their resilience and capacity to minimise overfitting. Preprocessing techniques like SMOTE, ADASYN, and K-Means clustering are used to prepare data for training and evaluation.

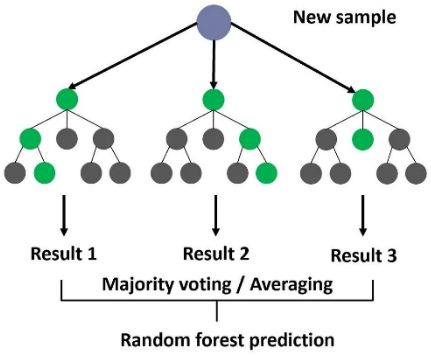


Figure 5: Random Forrest Algorithm (Source: TowardsDataSScience)

__**Support Vector Machine:**__ A supervised learning model called Support Vector Machine (SVM) (**Evgeniou, T. and Pontil, M., 2001**) looks for the feature space hyperplane that best splits the classes. SVM is capable of handling both linear and non-linear classifi-cation jobs thanks to its robust approach. SVM with a radial basis function (RBF) kernel is frequently used in the field of ransomware and malware detection because it makes it possible to design non- linear decision boundaries, which can

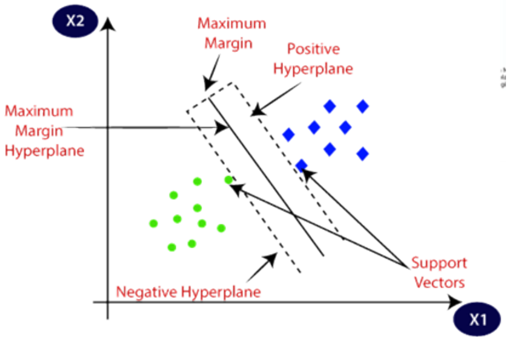


Figure 6: SVM (**Evgeniou, T. and Pontil, M., 2001**)

effectively capture intricate patterns in the data. SVM models are trained and assessed using preprocessed data produced with ADASYN, K-Means, and SMOTE clustering techniques. When these preprocessing techniques are incorporated into the model train- ing pipeline, SVM models perform better in malware detection tasks by managing class imbalances and capturing complex, non-linear correlations between features. The overall dependability and ef- fectiveness of malware detection systems are increased by this method, which also decreases false positives and improves the model's capacity to accurately categorise harmful software.

K – Nearest Neighbour (KNN): A straightforward but efficient classification technique called K-Nearest Neighbours (KNN)(**Guo, G., Wang, H., Bell, D. and Bi, Y., 2004**) groups data points according to the categorization of their neighbours. KNN can be used in conjunction with oversampling methods like SMOTE, ADASYN, and K-Means clustering to overcome class imbalance in ransomware and malware detection. KNN models can predict outcomes more accurately if the minority class (suspicious case behaviours) is adequately represented. By incorporating these preprocess- ing techniques, KNN is better able to identify ransomware and malware by recognising the complex patterns and similarities in the data.

Logistic Regression: A basic machine learning classification technique called logistic regression models a binary depen- dent variable by using the logistic function (**Peng, J., Lee, K. and Ingersoll, G., 2002**). SMOTE, ADASYN, and K-Means clustering prepro- cessed datasets can be used to train logistic regression for ransomware and malware detection. By improving the minority class's representation, these preprocessing methods help the Logistic Re- gression model better understand the line that separates Normalfrom malevolent behaviour. A more dependable and efficient malware and ransomware detection system is the end result.

LSTM: The Long Short-Term Memory (LSTM) (**Hochreiter, S. and Schmidhuber, J., 1997**) type of recurrent neural networks can figure out how order affects sequence prediction problems. When it comes to jobs that involve straight data, they work very quickly. In order to find ransomware and other malware, LSTM networks can be taught on activity patterns or system calls. Using preprocessing techniques like SMOTE, ADASYN, and K- Means clustering makes sure that the LSTM network has a balanced dataset, which makes it better able to learn from both good and bad patterns. This makes the detection model more effective and reliable.

3.3 Evaluation Metrics

Precision: The accuracy rate of the model's positive predictions relative to its total positive predictions is calledprecision (**Goutte, C. and Gaussier, E., 2005**). The ability of the model to avoid incorrectly labelling harmless software as harmful is what we mean when we talk about precision in ransomware and malware detection. If the model's accuracy in labelling software instances as harmful is high, then it is generally accurate.

Recall: Sometimes referred to as sensitivity or true positive rate, recall is the proportion of true positive predictions among all real positive cases in the dataset. Recall in ransomware and malware detection gauges the model's ability to find every instance of hostile software. A high recall shows that the model correctly detects most of the malware events in the dataset. Recall has mathematical form as follows:

F1-score: Considered as the harmonic mean of recall and accuracy, the F1-score is a single statistic that strikes a balance. Its value falls between 0 and 1; a greater number indicates improved model precision and recall. When the dataset exhibits an imbalance between positive and negative cases, the F1-score is especially helpful.

Malicious Detection App

Predict

☑ Not Malicious!!!

Malicious Detection App

Predict

☒ Malicious!!!!

Here are the screenshots for my results.

Chapter 4: Implementation

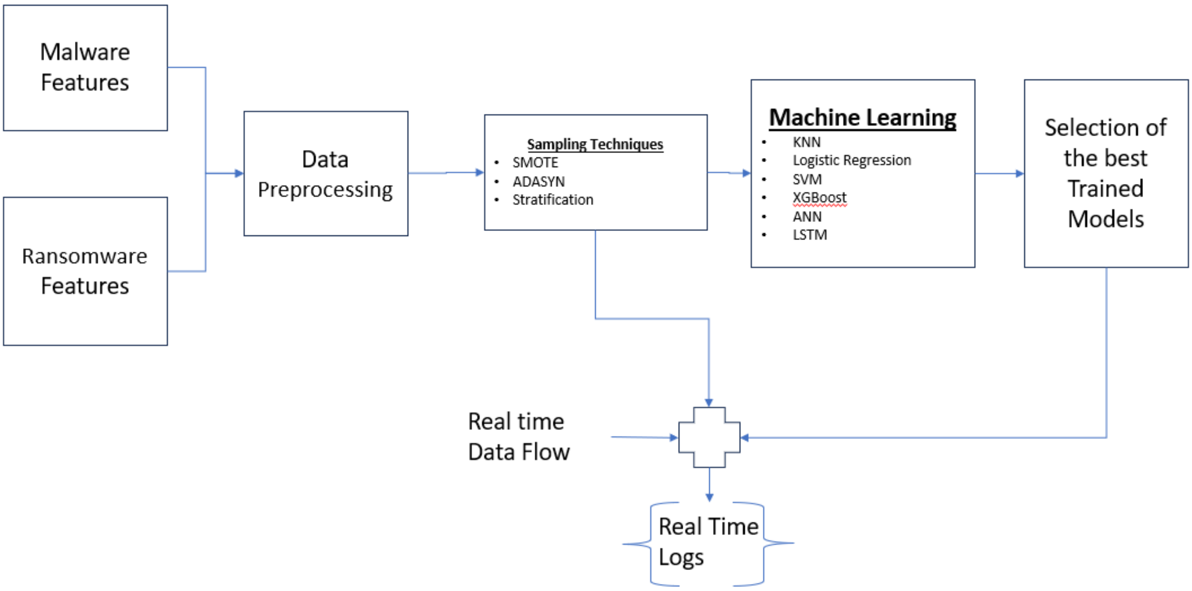


Figure 7: Implementation Framework

Step 1: Data Acquisition: In this step we have taken two important kinds of datasets for the security detection. In this we have taken Malware data and Ransomware data.

Step 2: Data Processing: In this step the data is either cleaned, or statistically corrected like outliers detection and cleaning or missing value imputation. So that data is made in a proper form to be modelled.

Step 3: Sampling: In this we have sampled the data in holdout method and then the data imbalance techniques are put in

Step 4: Modelling: Machine Learning models are fitted and trained in the training samples and checked in the testing samples to come out with a best performing trained pickle file

Step 5: Evaluation: The best model is selected through the greedy selection of the model using the accuracy, precision, recall, and f1 scores

<u>Step 6: Real Time monitoring</u>: In the real time scenario, the system monitors it the real time data response tracking, when the suspicious data is detected using the AI model that kind of solution is deployed and a log file is generated.

<u>Step 7: Log Generation</u>: The log files are sent to the security officers to take action.

Output: The solution that we have made is a response system. When this system is deployed into any kind of environment it will help in monitoring the system. The solution can be integrated into an API using th FLASK and can be deployed to any server like AWS. In that the API can be integrated to any enterprise server. This response system monitors in such a way that the solution can generate the log files of the intrusions or suspicious activity in th form of malware or ransomware.

4.2 Data Description

4.2.1 Ransomware

The dataset includes 62,485 cases and 18 variables; there are no missing cases, and all records are complete. These are identifiers and various numerical characteristics such as `FileName` and `md5Hash`, `Machine`, `DebugSize`, `ExportRVA` and others. The `Normal` column, which identifies if it is Normal or the contrary, also is consistent in a range, around the 43%. 4% of the files that they were able to categorize under Normal. This point is illustrated by the fact that values of `Machine`, `MajorImageVersion`, and `DllCharacteristics` are numerous and differ significantly, so we can suppose that dataset is rather diverse. For instance, `MajorImageVersion` has large standard deviations, meaning even though the greater part of the values are low, they may be compared to very high values.

4.2.2 Malware

The used dataset is specifically regarding malware detection, and includes 200000 entries of various attributes of network connections. Every entry contains knowledge about the time and type of connection events with references to the source and the destination connection, that is, the IP addresses and the connection ports and the networks protocols. The data-set also exhibits the services related to the connection, the time for which the connection has been active and the amount of the traffic that has been transmitted in both the directions.

4.3 Data Preprocessing

The data preprocessing that was done on the ransomware and malware datasets entailed some processes that are outlined below: The pre-processing steps involved first were data loading followed by data pre-processing through which unnecessary features were dropped from the modelling process. The observed missing values therefore required imputing by substituting the missing values by zeros in order to make the dataset to be complete. The cleaning started by the elimination of some attributes that had null values not easy to handle thus ensuring that all data used was clean data. Discrete data were decomposed into features that have two values in the form of one-hot encoding method in which the original categorical variables are omitted.

4.4 Data Preparation

Data pre-processing section in both the malware as well as the ransomware data set includes handling of class imbalance. In the first step, **SMOTE and ADASYN** methods was used to synthesize new samples from the minority classes and thereby balance the datasets. This made sure that the models could feed on a more diverse dataset and be able to learn on it. However, in the cases when **SMOTE and ADASYN** were not used, the data was split with an attempt to **stratify the resulting sets**. It kept class distribution constant between the training and testing sets which reduced the impact of imbalance issue but without creating synthetic samples

4.5 Modelling

4.5.1 Machine Learning

To accommodate ransomware detection, five various types of MSI models were used in order to assess the extent of various algorithmic solutions. These are a Random Forest classifier that educates numerous decision trees to lessen overfitting and achieve more exact results while going through the information; and a Support Vector Machine with correlation type (SVM-Lineal), which shows high outcomes in the sketch high dimensionality. Logistic Regression was used for its ease of interpretability and effectiveness for especially binary classification. Also, the K-Nearest Neighbors (KNN) classifier was adopted using proximity-sourced learning, while the XGBoost was also adopted because of its gradient-boosting and superior performance to large datasets and predictive tasks. Within the evaluation of each model, the parameters of the models were set to enhanced the performances for the ransomware dataset, including the number of estimators in Random Forest and number of neighbors in KNN models.

For malware detection, a similar approach was taken, involving the use of four machine learning models: Logistic Regression, Random Forest, K-Nearest Neigbors, XGBoost. The motivation for selecting Logistic Regression was based on the fact that it would perform a binary classification task and would allow a comparison with other complex models based on performance. For the third time, Random Forest was applied because of the ensemble approach, which increases the model's robustness and maintains its accuracy. K-Nearest Neighbors were used in order to perform model evaluation based on the proximity with respect to a determined metric in the feature space. XGBoost, that has a gradient boosting, was adopted because of its efficiency in working with and high accurate results, with handling imbalanced datasets. Every model was fine-tuned based on some hyperparameters; for instance, the maximum numbers of iterations for Logistic Regression as well as the number of features that should be used when implementing the XGBoost technique along with the proper evaluation metrics that was suitable for the malware dataset.

4.5.2 Deep Learning

For ransomware detection, two advanced machine learning models were employed: These two are a feedforward Artificial Neural Network (ANN) and a Long Short-Term Memory (LSTM) network. In developing the ANN model, an Multi-Layer Perceptron (MLP) with single layer of hidden neurons 100Neurons is used based on 300 iteration. The LSTM model was used to extract temporal information within the data and consisted of two LSTM layers with 50 nodes each together with dropout layers in between to avoid overfitting. The model used only one network layer with sigmoid activation function that was trained via Adam optimizer from the binary cross entropy loss. These models were intended to capture both the other elemental features and the temporal relationships in the data so as to improve the chances of identifying ransomware.

For malware detection, a similar method was employed – with the use of an ANN, as well as an LSTM model; however, some changes were made to meet the requirements of the used malware dataset. The last structure of the ANN model has two dense layers with ReLU activation functions in the first layer and SoftMax output layer for multi-class classification strategy. The LSTM model architecture in this paper had 64 and 32 nodes in the LSTM layers and the dropout layer was also used to avoid overfitting of the model. This model was compiled using Adam optimizer and categorical cross entropy loss after which training was done for 10 epochs using a batch size of 32. Both of these models were developed for targeting the complex structures that exist within the data, for the desired purpose of enhancing the accuracy and reliability of the malware identification process.

Chapter 5: Results

5.1 Malware

5.1.1 Machine Learning

5.1.1.2 Random Forest Classifier

Table 2: Random Forest Analysis for samplings

Technique	Accuracy	F1 Score (Normal)	F1 Score (Malicious)	Precision (Normal)	Precision (Malicious)	Recall (Normal)	Recall (Malicious)
ADASYN	0.917	0.91	0.923	0.992	0.862	0.841	0.993
SMOTE	0.953	0.951	0.956	1	0.915	0.907	1
Stratified	0.956	0.95	0.961	1	0.926	0.904	1

The Stratified Split technique demonstrated good accuracy of 0. 956 and relatively high F1 scores in all of them: 0. 961 for Suspicious or malicious case and 0. 95 for Normal labels. The same observation can be made with SMOTE technique though it has lower accuracy 0.953 to the Stratified Split, yet it has better F1 score for Suspicious case 0.951, hence indicating the technique's ability to support the generation of synthetic samples that improves the minority classes' performance. On the other hand, using ADASYN, the accuracy was the lowest 0.917; yet, it offered comparable F1 scores for both classes on par with or which surpassed most previous studies. This inflicts that although, ADASYN has the benefit of better sampling of the minority class, it is likely to add some amount of noise or overfitting, thus being inclined towards accuracy. Therefore, to conclude, all the techniques affect the model with a difference SMA and specifically Stratified Split and SMOTE looks balanced to deliver the accuracies with constant classification performance comparing to comparison actor ADASYN.

5.1.1.2 Logistic Regression

Table 3: Logistic Regression Analysis for samplings

Technique	Accuracy	F1 Score (Normal)	F1 Score (Malicious)	Precision (Normal)	Precision (Malicious)	Recall (Normal)	Recall (Malicious)
ADASYN	0.883	0.868	0.896	0.999	0.812	0.767	0.999
SMOTE	0.949	0.946	0.951	0.999	0.908	0.899	0.999
Stratified	0.953	0.946	0.959	0.999	0.922	0.899	0.999

A slightly different picture is paints out by the Stratified Split technique where the maximum accuracy of 0. 953 and overall had a good F1 score. There are 0.959 rate for the Suspicious case and 0. 946 for Normal classes. The SMOTE, proved to be slightly less accurate and that was at an accuracy of 0.949 but provided better F1 scores of the model for the Suspicious case instances (F1-Score = 0. 951), and hence it also demonstrates its suitability in handling class imbalance where the objective is to improve the classification of the minority class. On the other hand, by applying the ADASYN technique we obtained the lowest accuracy which was 0. 883. This means that although ADASYN is capable of boosting up the minority class, it also brings about variation that impacts the performance. In general, group 2 has more balanced accuracy measures, where Stratified Split and SMOTE perform better than others, but ADASYN has lower accuracy but produces more synthetic data to consider.

5.1.1.4 KNN

Table 4: K-NN Analysis for samplings

Technique	Accuracy	F1 Score (Normal)	F1 Score (Malicious)	Precision (Normal)	Precision (Malicious)	Recall (Normal)	Recall (Malicious)
ADASYN	0.888	0.888	0.887	0.885	0.891	0.891	0.884

SMOTE	0.942	0.94	0.944	0.975	0.914	0.908	0.977
Stratified	0.948	0.941	0.954	0.982	0.924	0.903	0.986

As for the K-Nearest Neighbours (KNN), the evaluation criteria show that the best accuracy of 0.948, along with balanced F1 Scores of 0.982 for Normal and 0.954 for Malicious. This shows that the Stratified Split approach is efficient in managing both classes of students. The SMOTE technique has an accuracy of 0. 942, enhanced the F1 Score for Suspicious case cases to 0.944 and sustained high accuracy for the Suspicious case circumstance at 0.942. On the other hand, ADASYN obtained the lowest accuracy of only 0. 88 while it had a specificity of 0.888 on Normal cases it had a good precision of 0.887.

5.1.1.5 XGBoost

Table 5: XGBoost Analysis for samplings

Technique	**Accuracy**	**F1 Score (Normal)**	**F1 Score (Malicious)**	**Precision (Normal)**	**Precision (Malicious)**	**Recall (Normal)**	**Recall (Malicious)**
ADASYN	0.889	0.876	0.9	0.998	0.82	0.78	0.998
SMOTE	0.952	0.95	0.954	1	0.913	0.905	1
Stratified Split	0.956	0.949	0.961	1	0.926	0.904	1

For XGBoost, the results show that the Stratified Split technique achieved the highest accuracy of 0.956, with impressive F1 Scores of 0.949 for Normal and 0.961 for Malicious, indicating a well-balanced performance across both classes. The SMOTE technique also demonstrated strong results with an accuracy of 0.952 and high F1 Scores of 0.95 for Normal and 0.954 for Malicious, reflecting its effectiveness in managing class imbalance. ADASYN, while achieving a slightly lower accuracy of 0.889, still provided a high F1 Score for Suspicious case cases at 0.9 and maintained strong precision for Normal instances at 0.998, suggesting good performance in enhancing minority class representation.

5.1.2 Deep Learning

5.1.2.1 ANN

The ANN model was evaluated using three techniques: SMOTE, ADASYN, and Stratified Split. The model achieved a high accuracy across all techniques, with the highest accuracy of 95.62% using Stratified Split and SMOTE. Precision for *Normal* instances was near-perfect across all techniques, consistently around 99.9%. For suspicious case instances, the precision was slightly lower, ranging from 91.2% to 92.6%. The model demonstrated strong recall for *Normal* instances (around 99.9%) and good recall for suspicious case instances, though slightly lower. F1 scores followed a similar pattern, indicating balanced performance across all metrics with the Stratified Split technique yielding the best overall results.

Table 6: XGBoost Analysis for samplings

Technique	**Accuracy**	**Precision (Normal)**	**Precision (Malicious)**	**Recall (Normal)**	**Recall (Malicious)**	**F1 Score (Normal)**	**F1 Score (Malicious)**
SMOTE	0.956076	0.999232	0.919787	0.912852	0.999299	0.954091	0.957895
ADASYN	0.94944	0.99919	0.91272	0.894176	0.999345	0.94377	0.954071
Stratified Split	0.956199	0.999877	0.925564	0.904045	0.999906	0.949549	0.9613

5.1.2.2 LSTM

For the LSTM model, three techniques—SMOTE, ADASYN, and Stratified Split—were employed to handle class imbalance. The Stratified Split technique yielded the highest accuracy at 95.62%, closely followed by SMOTE at 95.63%. ADASYN had a slightly lower accuracy of 95.01%. In terms of precision for the "Normal" class, all three techniques performed exceptionally well, with values close to 100%. Precision for the "Malicious" class was highest with Stratified Split at 92.55%, while ADASYN and SMOTE followed with 91.38% and 92.02%, respectively. The recall for the "Normal" class remained high across all techniques, but for the "Malicious" class, SMOTE performed best with 95.81%, followed by Stratified Split at 96.13%, and ADASYN at 95.46%. Overall, the LSTM model demonstrated strong and consistent performance across different balancing techniques.

Table 7: XGBoost Analysis for samplings

Technique	**Accuracy**	**Precision (Normal)**	**Precision (Malicious)**	**Recall (Normal)**	**Recall (Malicious)**	**F1 Score (Normal)**	**F1 Score (Malicious)**
SMOTE	0.956309	0.999233	0.920183	0.91332	0.999299	0.954347	0.95811
ADASYN	0.950103	0.999191	0.913774	0.895573	0.999345	0.944549	0.954646
Stratified Split	0.956173	0.999877	0.925524	0.903989	0.999906	0.949518	0.961278

5.1.3 Discussion and Analysis

The accuracy comparison chart also reveals that ANN and LSTM models are superior to other algorithms with the accuracy of 0. 954. This shows their suitability in dealing with the given dataset as will be shown later on in this paper. Random Forest and XGBoost also shown good results with accuracy values of 0. 942 and 0. 932, respectively. However, as for K-Nearest Neighbours and Logistic Regression, the accuracy is somewhat lower, where K-Nearest Neighbours was 0. 926, and Logistic Regression at 0. 928. This means that all the models are good but deep learning models like ANN and LSTM and ensemble models like Random Forest are the best for this task.

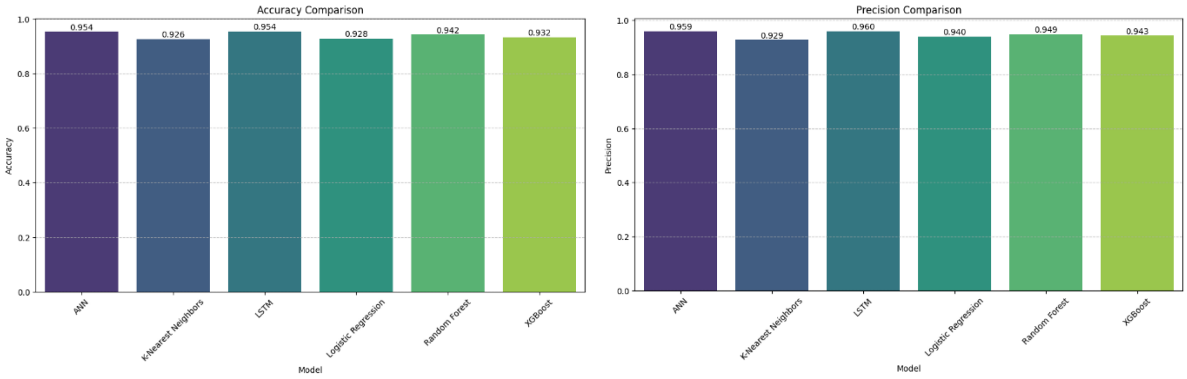


Figure 8: Accuracy and Precision comparison of all the models

In the precision comparison chart, the LSTM model has the highest score of 0. 960, which is very near to ANN with 0. 959. This implies that these models are very useful in reducing the number of false positives. Random Forest also gives a satisfactory result with precision of 0. 949, which is quite high and thus, the model can be relied on for prediction tasks. XGBoost and Logistic Regression models have precision scores equal to 0. 943 and 0. 940, respectively, which are a little lower but still good figures. K-Nearest Neighbors, with a level of accuracy of 0. 929, is lower than the other models, suggesting that it might be slightly higher in false positive rate in this regard. In general, LSTM and ANN models show better accuracy and can be considered as highly accurate models for prediction.

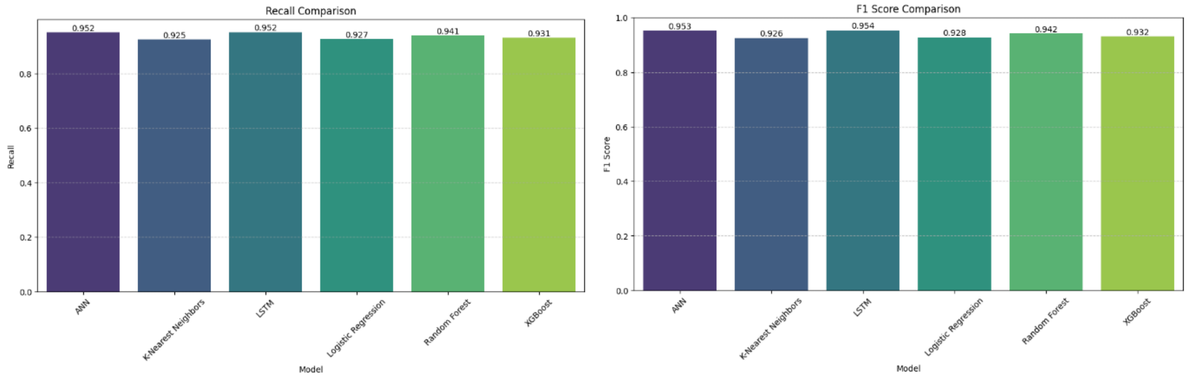


Figure 9: Recall and F1 Score comparison of different models

From recall comparison chart, it is also shown that both the ANN and LSTM models have the highest recall value of 0. 952, which shows that they are very good at identifying positive cases. This means that these models are less likely to fail to identify actual positive cases, which are important in cases where all positive cases must be identified. Random Forest comes second with a recall score of 0. 941, which proves the ability of the model to show good results in identifying positives, although it is somewhat slower than ANN and LSTM. XGBoost and Logistic Regression have recall scores of 0. 931 and 0. 927, respectively, which are still low but not as low as the previous one, and still comparatively competitive. The lowest recall is recorded by K-Nearest Neighbors at 0. 925, thereby making it capable of missing more positive cases than the other models. In general, ANN and LSTM models are the most accurate in terms of the highest recall, which is useful when it is necessary to detect all positive cases.

The F1 scores of the models provide much information about the models' performance. In the case of F1 score, the LSTM model emerges with the highest score of 0. 954, which shows the capacity of the algorithm to achieve high levels of both precision and recall at the same time and, therefore, it is the most suitable one to minimize both the number of false positives and false negatives. The Artificial Neural Network (ANN) is in the second place with accuracy of 0. 953, which also shows good results in classification problems. Logistic Regression, with F1 score of 0. 928, has a good but slightly lower accuracy compared to ANN and LSTM. K-Nearest Neighbors (KNN) has the least F1 score of 0. 926 and that is why it is less balanced in terms of precision/Recall ratio and might have more potential classification errors. This is well illustrated in the bar graph where LSTM and ANN outperform the other models which are evident from the bar graph.

5.2 Ransomware

5.2.1 Machine Learning

5.2.1.1 Random Forest Classifier

The models achieved the following metrics: Accuracy ranges from 0.9950 to 0.9970. The F1 Score for Normal instances is sandwiched between 0.9956 and 0.9973, while for suspicious case instances, it ranges from 0.9942 to 0.9965. Precision for Normal instances vary from 0.9935 to 0.9971, and for suspicious case instances, it ranges from 0.9952 to 0.9970. Recall for Normal instances is between 0.9977 and 0.9915, and for suspicious case instances, it ranges from 0.9961 to 0.9955.

Table 8: Random Forest Analysis for samplings

Technique	**Accuracy**	**F1 Score (Normal)**	**F1 Score (Malicious)**	**Precision (Normal)**	**Precision (Malicious)**	**Recall (Normal)**	**Recall (Malicious)**
ADASYN	0.9950	0.9956	0.9942	0.9935	0.9970	0.9977	0.9915
SMOTE	0.9970	0.9973	0.9965	0.9971	0.9968	0.9973	0.9961

Stratified Split	0.9960	0.9965	0.9953	0.9966	0.9952	0.9964	0.9955

5.2.1.2 SVM

For the SVM model, engaging the ADASYN technique resulted in an accuracy of 0.8656, with F1 Scores of 0.89 for Normal instances and 0.83 for suspicious case instances. The precision was 0.83 for Normal and 0.920 for suspicious case instances, with recall values of 0.95 for Normal and 0.75 for malicious. The SMOTE technique produced an accuracy of 0.8851, with F1 Scores of 0.90 for Normal and 0.87 for suspicious case instances. Precision was 0.93 for Normal and 0.83 for malicious, and recall was 0.86 for Normal and 0.91 for suspicious case instances. Lastly, the Stratified Split technique achieved an accuracy of 0.8954, with F1 Scores of 0.91 for Normal and 0.88 for suspicious case instances. Precision values were 0.93 for Normal and 0.85 for malicious, with recall values of 0.88 for Normal and 0.92 for suspicious case instances.

Table 9: SVM Analysis for samplings

Technique	Accuracy	F1 Score (Normal)	F1 Score (Malicious)	Precision (Normal)	Precision (Malicious)	Recall (Normal)	Recall (Malicious)
ADASYN	0.8656	0.89	0.83	0.83	0.920	0.95	0.75
SMOTE	0.8851	0.90	0.87	0.93	0.83	0.86	0.91
Stratified Split	0.8954	0.91	0.88	0.93	0.85	0.88	0.92

5.2.1.3 Logistic Regression

In the Logistic Regression model, the Stratified Split technique achieved the highest accuracy of 0.9920, with F1 Scores, precision, and recall all consistently at 0.99 for both Normal and Malicious case instances. The SMOTE technique resulted in a moderate accuracy of 0.8856, with F1 Scores of 0.90 for Normal and 0.87 for malicious, along with precision and recall values close to each other. ADASYN had the lowermost accuracy at 0.8373, with F1 Scores of 0.86 for Normal and 0.81 for malicious. It also showed lower precision and recall for suspicious case instances compared to the other techniques.

Table 10: Logistic Regression Analysis for samplings

Technique	Accuracy	F1 Score (Normal)	F1 Score (Malicious)	Precision (Normal)	Precision (Malicious)	Recall (Normal)	Recall (Malicious)
ADASYN	0.8373	0.86	0.81	0.86	0.81	0.85	0.80
SMOTE	0.8856	0.90	0.87	0.90	0.87	0.90	0.86
Stratified Split	0.9920	0.99	0.99	0.99	0.99	0.99	0.99

5.2.1.4 KNN

For the KNN model, the outcomes are as follows: Using ADASYN, the accuracy was 0.9875, with F1 Scores of 0.989 for Normal instances and 0.9856 for Malicious case instances. Precision was 0.982 for Normal and 0.9948 for malicious, with recall values of 0.996 for Normal and 0.9766 for malicious. The SMOTE technique enhanced the accuracy to 0.9912, with F1 Scores of 0.9923 for Normal and 0.9898 for suspicious case instances. Precision was 0.9909 for Normal and 0.9916 for malicious, with recall values of 0.9937 for Normal and 0.9879 for malicious. The Stratified Split technique resulted in an accuracy of 0.9920, with F1 Scores of 0.99 for Normal and 0.89 for suspicious case instances. Precision was 0.92 for Normal and 0.88 for malicious, with recall values of 0.90 for both Normal and suspicious case instances. The KNN model performed best with the SMOTE technique, achieving the highest accuracy and strong F1 scores. The Stratified Split technique showed significantly lower performance for malicious case instances, particularly in F1 Score, precision, and recall.

Table 11: k-NN Analysis for samplings

Technique	Accuracy	F1 Score (Normal)	F1 Score (Malicious)	Precision (Normal)	Precision (Malicious)	Recall (Normal)	Recall (Malicious)
ADASYN	0.9875	0.989	0.9856	0.982	0.9948	0.996	0.9766
SMOTE	0.9912	0.9923	0.9898	0.9909	0.9916	0.9937	0.9879
Stratified Split	0.9920	0.99	0.89	0.92	0.88	0.90	0.90

5.2.1.5 XGBoost

For the XGBoost model, the results are as follows: Using ADASYN, the accuracy was 0.9991, with F1 Scores of 0.9995 for Normal instances and 0.994 for suspicious case instances. Precision was 0.9993 for Normal and 0.9970 for malicious, with recall values of 0.9998 for Normal and 0.9909 for malicious. The SMOTE technique resulted in an accuracy of 0.9960, with F1 Scores of 0.9965 for Normal and 0.9953 for suspicious case instances. Precision and recall values for both Normal and suspicious case instances were identical at 0.9965 and 0.9953, respectively. The Stratified Split technique yielded a similar accuracy of 0.9958, with F1 Scores of 0.9964 for Normal and 0.9952 for suspicious case instances. Precision for Normal was 0.9965 and 0.9950 for malicious, with recall values of 0.9962 for Normal and 0.9953 for malicious. The XGBoost model performed best with the ADASYN technique, achieving the highest accuracy and superior recall for Normal instances. The SMOTE and Stratified Split techniques yielded very similar results, with only slight variations in F1 Scores and precision for the suspicious case instances.

Table 12: XGBoost Analysis for samplings

Technique	Accuracy	F1 Score (Normal)	F1 Score (Malicious)	Precision (Normal)	Precision (Malicious)	Recall (Normal)	Recall (Malicious)
ADASYN	0.9991	0.9995	0.994	0.9993	0.9970	0.9998	0.9909
SMOTE	0.9960	0.9965	0.9953	0.9965	0.9953	0.9965	0.9953
Stratified Split	0.9958	0.9964	0.9952	0.9965	0.9950	0.9962	0.9953

5.2.2 Deep Learning Techniques

5.2.2.1 ANN

For the ANN model, the results are as follows: Using ADASYN, the accuracy was 0.9815, with F1 Scores of 0.9836 for Normal instances and 0.9788 for suspicious case instances. Precision was 0.9735 for Normal and 0.9922 for malicious, with recall values of 0.9940 for Normal and 0.9657 for malicious. The SMOTE technique improved the accuracy to 0.9867, with F1 Scores of 0.9883 for Normal and 0.9846 for suspicious case instances. Precision was 0.9864 for Normal and 0.98723 for malicious, with recall values of 0.9903 for Normal and 0.982 for malicious. The Stratified Split technique resulted in the highest accuracy of 0.9874, with F1 Scores of 0.9889 for Normal and 0.9853 for suspicious case instances. Precision was 0.9872 for Normal and 0.9875 for malicious, with recall values of 0.9906 for Normal and 0.9831 for malicious. The ANN model performed best with the Stratified Split technique, achieving the highest accuracy and strong F1 scores. The ADASYN technique resulted in slightly lower performance, especially in recall for suspicious case instances.

Table 13: ANN Analysis for samplings

Technique	Accuracy	F1 Score (Normal)	F1 Score (Malicious)	Precision (Normal)	Precision (Malicious)	Recall (Normal)	Recall (Malicious)
ADASYN	0.9815	0.9836	0.9788	0.9735	0.9922	0.9940	0.9657
SMOTE	0.9867	0.9883	0.9846	0.9864	0.98723	0.9903	0.982

Stratified Split	0.9874	0.9889	0.9853	0.9872	0.9875	0.9906	0.9831

5.2.2.2 LSTM

The LSTM model achieved the following results: Using ADASYN, the accuracy was 0.9815, with an F1 Score of 0.9836 for Normal instances and 0.9788 for malicious case instances. Precision was 0.9735 for Normal and 0.9922 for malicious, with recall values of 0.9940 for Normal and 0.9657 for malicious. With SMOTE, the accuracy improved to 0.9867, F1 Scores were 0.9883 for Normal and 0.9846 for suspicious case instances. Precision was 0.9864 for Normal and 0.98723 for malicious, with recall values of 0.9903 for Normal and 0.982 for malicious. The Stratified Split technique resulted in an accuracy of 0.9770, with F1 Scores of 0.9889 for Normal and 0.9798 for malicious case instances. Precision was 0.9734 for Normal and 0.9777 for malicious, with recall values of 0.9830 for Normal and 0.9692 for malicious. The LSTM model performed best with the SMOTE technique across most metrics, including accuracy and F1 scores. The ADASYN technique yielded the lowest recall for suspicious case instances, indicating its weaker performance compared to the other techniques.

Table 12: LSTM Analysis for samplings

Technique	**Accuracy**	**F1 Score (Normal)**	**F1 Score (Malicious)**	**Precision (Normal)**	**Precision (Malicious)**	**Recall (Normal)**	**Recall (Malicious)**
ADASYN	0.9815	0.9836	0.9788	0.9735	0.9922	0.9940	0.9657
SMOTE	0.9867	0.9883	0.9846	0.9864	0.98723	0.9903	0.982
Stratified Split	0.9770	0.9889	0.9798	0.9734	0.9777	0.9830	0.9692

5.2.3 Discussion and Analysis

5.2.3.1 Accuracy and F1-score

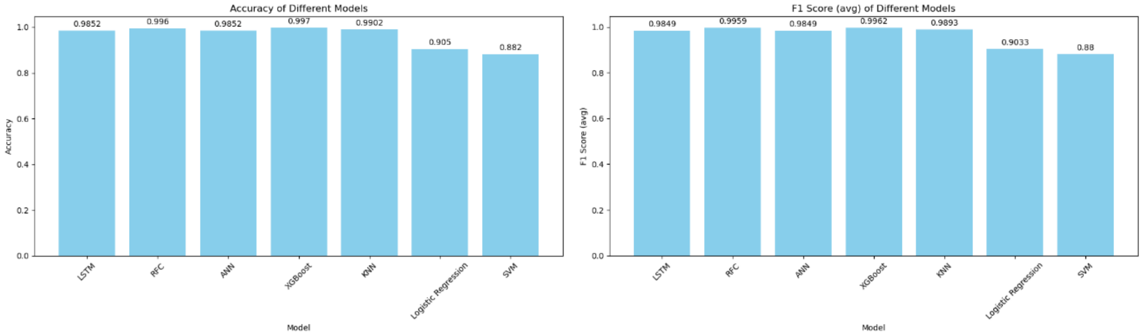


Figure 10: Accuracy and f1 score comparison

XGBoost leads with the highest accuracy of 99.70%, followed by RFC at 99.60%. Both models display superior performance in classifying instances correctly. KNN also shows a strong accuracy of 99.02%, demonstrating its effectiveness. LSTM and ANN achieve accuracies of 98.52%, showcasing their strong performance but slightly lower than XGBoost and RFC. Logistic Regression and SVM have the lowest accuracies at 90.50% and 88.20%, respectively, indicating they are less effective overall in classification. XGBoost leads with the highest accuracy of 99.70%, followed by RFC at 99.60%. Both models exhibit superior performance in classifying instances correctly. KNN also shows a strong accuracy of 99.02%, demonstrating its effectiveness. LSTM and ANN achieve accuracies of 98.52%, reflecting their strong performance but slightly lower than XGBoost and RFC. Logistic Regression and SVM have the lowest accuracies at 90.50% and 88.20%, respectively, indicating they are less effective overall in classification.

5.2.3.3 Precision and Recall

XGBoost leads in average precision with 99.66%, showing its effectiveness at minimizing false positives. RFC is slightly behind with an average precision of 99.60%. LSTM and ANN both have an average precision of 98.57%, which is strong but not as high as XGBoost and RFC. KNN has an average precision of 95.99%, which is lower compared to the top models. Logistic Regression and SVM have average precisions of 90.33% and 88.17%, respectively, indicating they struggle with precision.

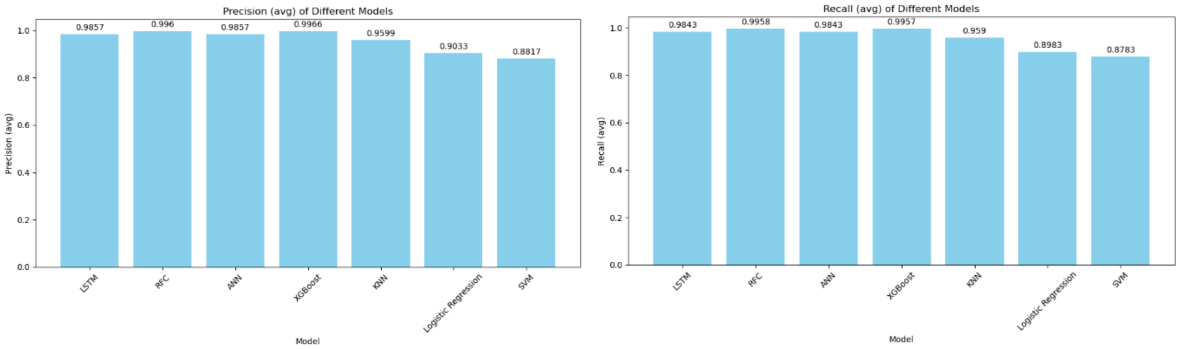


Figure 11: Precision and Recall comparison comparison

XGBoost excels in recall with 99.57%, capturing the majority of true positives. RFC has a very close recall of 99.58%. LSTM and ANN both have recall values of 98.43%, which are strong but slightly lower than XGBoost and RFC. KNN's recall is 95.90%, which is good but not as high as the leading models. Logistic Regression and SVM have the lowest recall values at 89.83% and 87.83%, respectively, suggesting they miss a higher proportion of true positives.

5.3 Summary

In case of ransomware, the performance comparison of several models reveals that both ANN and LSTM excel across numerous metrics. They each achieve an accuracy of 0.954, demonstrating robustness in handling the dataset. The LSTM model also leads in precision with a score of 0.960 and in recall with 0.952, making it highly effective at minimizing false positives and capturing all true positives. ANN follows closely with precision of 0.959 and recall of 0.952, showing similarly strong performance. Random Forest and XGBoost also perform well with accuracies of 0.942 and 0.932, respectively. However, K-Nearest Neighbors and Logistic Regression lag behind, with slightly lower accuracy, precision, recall, and F1 scores. The F1 scores further highlight LSTM as the top performer with 0.954, balancing precision and recall effectively, while ANN also demonstrates strong classification capability with a score of 0.953.

For malware also we get XGBoost and RFC models above in all the metrics and hence we can say these two are more reliable models for this task. They also perform fairly well although not as well as XGBoost and RFC. In fact, KNN which is among the best-performing classification algorithms has comparatively lower precision and recall results as those of other models. Logistic Regression and SVM give lower accuracy in all the aspects and it suggests that tuning of these models, or attempting different models would useful.

Chapter 6: Conclusion and Future Work

This work also shows the ability of the modern machine learning approaches in identification of ransomware and malware. The Random Forest model outperformed all the other models with high accuracy and F1-score. Other methods like K-Nearest Neighbors (KNN) and Support Vector Machine (SVM) also performed very well the accuracy of KNN was close to one while, using methods such as SMOTE and ADASYN, SVM also exhibited high level of accuracy. Other techniques, such as

Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) provided much higher accuracy, 99. As for instance, SMOTE achieved an average accuracy of 98% confirming its capacity for handling data despite having been synthesized for sequential pattern data. Logistic Regression was the most accurate with a percentage of 93.83%.

SMOTE, ADASYN were used to overcome the class imbalance problem as well as to improve the performance of models. Further, there was enhancement of feature representation and the models by the use of K-Means clustering. The paper points at the importance of complex machine learning algorithms post-processing steps in countering new forms of ransomware which continue to emerge. Through these advanced approaches integrated in the research, this will enhance the cybersecurity posture formidable to new challenging cyber threats.

The future work should then, centre its efforts in fine-tuning and advancement of the current machine learning algorithms the identification and detection of ransomware and malware more effectively. This includes the ability of real time threat detection, adaptiveness to incorporate such threats, and research not only to specific types of threats but also expand the scope to multiple datasets. Although there have been advancements in this sector in the recent past, there are still issues that need further enhancement before the models can go to the market or can be integrated with currently existing computer security systems and some of the most significant of these include the following.; Furthermore, the concerns of ethics and data protection will also be critically important in order to make the best use of these sophisticated methods in practice.

References

Ali, J., Khan, R., Ahmad, N. and Maqsood, I., 2012. Random forests and decision trees. International Journal of Computer Science Issues (IJCSI), 9.

Alraizza, A. and Algarni, A., 2023. Ransomware detection using machine learning: A survey. Big Data and Cognitive Computing, 7(3), p.143.

Bae, S., Lee, G. and Im, E.G., 2019. Ransomware detection using machine learning algorithms. Concurrency and Computation: Practice and Experience, 32, e5422.

Buriro, A., Buriro, A.B., Ahmad, T., Buriro, S. and Ullah, S., 2023. Malwd&c: A quick and accurate machine learning-based approach for malware detection and categorization. Applied Sciences, 13(4), p.2508.

Charmilisri, A., Harshi, I., Madhushalini, V. and Raja, L., 2023. A novel ransomware virus detection technique using machine and deep learning methods. In 2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS), pp.8-14. IEEE.

Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P., 2002. SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research (JAIR), 16, pp.321-357.

Chesti, I.A., Humayun, M., Sama, N.U. and Jhanjhi, N.Z., 2020. Evolution, mitigation, and prevention of ransomware. In 2020 2nd International Conference on Computer and Information Sciences (ICCIS), pp.1-6.

Choudhary, S. and Sharma, A., 2020. Malware detection classification using machine learning. pp.1-4.

Conti, M., Dehghantanha, A., Franke, K. and Watson, S., 2018. Internet of things security and forensics: Challenges and opportunities. Future Generation Computer Systems, 78, pp.544-546.

Evgeniou, T. and Pontil, M., 2001. Support vector machines: Theory and applications. In Machine Learning: Theory and Applications, pp.249-257.

Goutte, C. and Gaussier, E., 2005. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In European Conference on Information Retrieval (ECIR), pp.345-359.

Guo, G., Wang, H., Bell, D. and Bi, Y., 2004. KNN model-based approach in classification.

Hammadeh, K. and Kavitha, M., 2023. Unraveling ransomware: Detecting threats with advanced machine learning algorithms. International Journal of Advanced Computer Science and Applications, 14(9).

He, H., Bai, Y., Garcia, E. and Li, S., 2008. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. pp.1322-1328.

Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. Neural Computation, 9, pp.1735-1780.

Humayun, M., Jhanjhi, N.Z., Alsayat, A. and Ponnusamy, V., 2021. Internet of things and ransomware: Evolution, mitigation and prevention. Egyptian Informatics Journal, 22(1), pp.105-117.

Kamboj, A., Kumar, P., Bairwa, A.K. and Joshi, S., 2023. Detection of malware in downloaded files using various machine learning models. Egyptian Informatics Journal, 24(1), pp.81-94.

Kharraz, A., Arshad, S., Mulliner, C., Robertson, W. and Kirda, E., 2016. Unveil: A large-scale, automated approach to detecting ransomware.

Kouliaridis, V., Barmpatsalou, K., Kambourakis, G. and Chen, S., 2020. A survey on mobile malware detection techniques. IEICE Transactions on Information and Systems, E103-D, pp.204-211.

Li, Y. and Wu, H., 2012. A clustering method based on k-means algorithm. Physics Procedia, 25, pp.1104-1109.

Majd, N.E.M. and Mazumdar, T., 2023. Ransomware classification using machine learning. In 2023 32nd International Conference on Computer Communications and Networks (ICCCN), pp.1-7. IEEE.

Mkandawire, Y. and Zimba, A., 2023. A supervised machine learning ransomware host-based detection framework. Zambia ICT Journal, 7(1), pp.52-56.

Mohammed, M.A., Lakhan, A., Zebari, D.A., Abdulkareem, K.H., Nedoma, J., Martinek, R., Tariq, U., Alhaisoni, M. and Tiwari, P., 2023. Adaptive secure malware efficient machine learning algorithm for healthcare data. CAAI Transactions on Intelligence Technology.

Muniandy, M., Ismail, N., Al-Nahari, A. and Ngo Yao, D., 2024. Evolution and impact of ransomware: Patterns, prevention, and recommendations for organizational resilience. International Journal of Academic Research in Business and Social Sciences, 14.

Peng, J., Lee, K. and Ingersoll, G., 2002. An introduction to logistic regression analysis and reporting. Journal of Educational Research, 96, pp.3-14.

Rahman, M.S., Sabbir, M.S.A. and Ghosh, S., 2024. Ransomware attack detection using machine learning approaches. In 2024 3rd International Conference for Innovation in Technology (INOCON), pp.1-7. IEEE.

Sood, A.K. and Enbody, R.J., 2013. Targeted cyberattacks: A superset of advanced persistent threats. IEEE Security & Privacy, 11(1), pp.54-61.

Suarez-Tangil, G., Tapiador, J., Peris-Lopez, P. and Ribagorda, A., 2013. Evolution, detection and analysis of malware for smart devices. IEEE Communications Surveys & Tutorials, 16.

Wadho, S.A., Yichiet, A., Gan, M.L., Lee, C.K., Ali, S. and Akbar, R., 2024. Ransomware detection techniques using machine learning methods. In 2024 IEEE 1st Karachi Section Humanitarian Technology Conference (KHI-HTC), pp.1-6. IEEE.