Machine Learning-Based Crypto Ransomware Detection Model On Windows Platforms

Martin Musonda[a], Aaron Zimba[b], Sinyinda muwanei[c]
a. Department of Computer Science and Information Technology, Mulungushi University Kabwe, Zambia
mdkmusonda@gmail.com
b. Department of Computer Science, ZCAS University, Lusaka Zambia, email: aaron.zimba@zcasu.edu.zm

c. Department of Computer Science and Information Technology, Mulungushi University Kabwe, Zambia
msiniyinda@mu.edu.zm

***Abstract*— Ransomware, an evolving and highly destructive form of malware, presents substantial challenges in terms of detection and prevention. Despite extensive research and the application of Machine Learning (ML) models, existing defense mechanisms have struggled to provide complete protection, as most ML models fall short of achieving perfect detection rates. The study aimed to achieve several objectives related to Crypto-Ransomware detection. Firstly, it involved an examination of current ML frameworks employed in this field and the identification of associated challenges. Subsequently, the study focused on the creation of a new machine learning model designed for the detection and analysis of Crypto-Ransomware. By capitalizing on the shared behavioral patterns exhibited by ransomware, the proposed model attains an impressive 98% accuracy in recognizing ransomware on Windows systems. Lastly, the developed model's effectiveness in identifying Crypto-Ransomware was assessed through validation processes. Through evaluating multiple classifiers, the study identifies the Random Forest algorithm as the optimal choice for the model. This research marks a notable advancement in robust ransomware detection, working towards mitigating the far-reaching impacts of Crypto ransomware, a pervasive cyber threat.**

***Keywords*— Crypto ransomware, Machine learning**

I. Introduction

In today's digital landscape, cybercrime has transformed into a profitable enterprise[1], with malicious agents exploiting the internet to carry out nefarious activities[2]. Among the various forms of cyber threats, ransomware has emerged as a significant concern[3], causing substantial financial losses and operational disruptions for individuals and organizations worldwide. Ransomware is a type of malware that encrypts data or restricts device functionality until a ransom is paid by the victim [4]. Despite advancements in software and hardware security, traditional antivirus solutions struggle to keep up with the rapidly evolving ransomware landscape. This has prompted the exploration of advanced techniques such as machine learning for effective detection and mitigation.

Unlike other types of malware, the impact of ransomware is irreversible. Even after neutralizing an attack, encrypted files remain locked until a decryption key is obtained [5][6]. With the continuous development of technology, ransomware has evolved, exploiting improved computing resources and advanced cryptographic operations. The rise of automation has further facilitated criminal schemes, with ransomware becoming an increasingly harmful tool in the hands of attackers [7][8].

Ransomware comes in various categories, including Crypto ransomware, Locker ransomware, Hybrid ransomware (combining Locker and Crypto attacks), and Scareware [6]. Among these, Crypto ransomware poses a significant threat as it encrypts a victim's data files, rendering them inaccessible without the decryption key [9]. This article focuses primarily on Crypto ransomware and aims to develop a machine learning-based detection model to mitigate attacks on the widely used Windows operating system.

This research explores the application of machine learning in ransomware detection, investigating its potential, challenges, and promising approaches. As ransomware incidents continue to escalate [10][11] and impact critical sectors like health[12] and education, there is an urgent need to develop robust and effective methods [13][14]for detecting and combating this pervasive cyber threat. By leveraging the power of machine learning[15], we can strive towards enhancing cybersecurity measures and minimizing the detrimental impacts of ransomware attacks.

II. Background

Ransomware, a pervasive cyber threat, has impacted industries across the board, targeting small businesses, large enterprises, banking institutions, and medical organizations [16][15]. While some victims who pay the ransom may regain access to their data, there is no guarantee of future protection from the same attackers. The number of ransomware attacks has steadily increased in recent years, surpassing the figures of the previous decade [3].

A typical ransomware attack consists of four stages: infection or deployment, encryption/locking, demand, and result [17]. The initial infection stage can occur through various methods, including phishing, social engineering, accessing infected websites, or exploiting vulnerabilities in applications and drivers [16][17][18]. Once a victim's device is infected, the ransomware enters the encryption and locking stage. Crypto ransomware utilizes either Private Key or public key.[18]

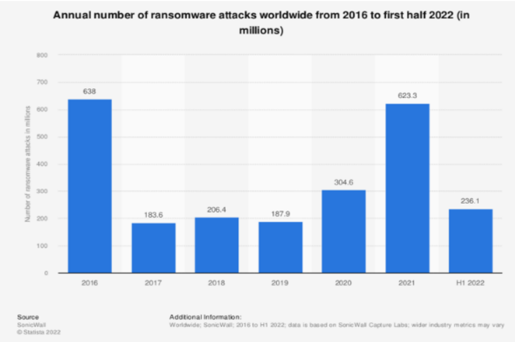


Fig. 1. Number of ransomware attacks per year 2016-H1 2022.

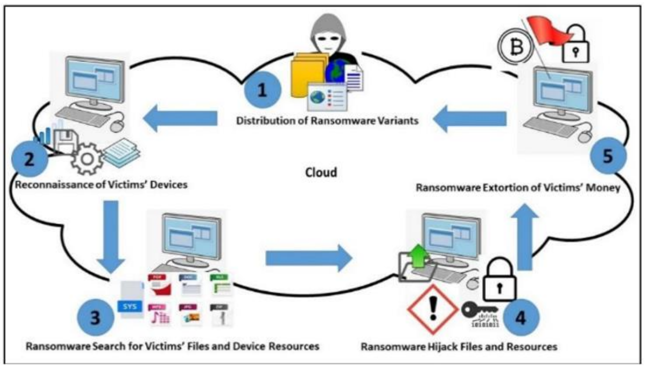

1
Distribution of Ransomware Variants
2
Reconnaissance of Victims' Devices
Cloud
5
Ransomware Extortion of Victims' Money
3
Ransomware Search for Victims' Files and Device Resources
4
Ransomware Hijack Files and Resources

Fig. 2. A ransomware routine[18].

Cryptosystem (PrCR) or Public Key Cryptosystem (PuCR), generating cryptographic keys to encrypt selected file types [19]. The attackers may remotely generate the keys on a command-and-control (C2C) server, which also acts as a repository for decryption keys after the ransom is paid [20].

In the demand stage, the victim receives a notification detailing the ransom payment instructions and timeframe. The outcome of an attack depends on the victim's actions, including paying the ransom or attempting to recover data from unaffected backups. Ransomware poses a significant threat not only to systems but also to individuals, as seen in the escalating attacks on medical institutions [21].

Understanding the infection vectors of ransomware[19] is crucial in developing effective mitigation strategies. Common vulnerabilities leading to ransomware delivery include phishing scams, poor user practices, weak passwords, malicious websites, stolen user credentials, and a reluctance to adopt security solutions [16].

Ransomware stands apart from other malware categories due to its distinct characteristic of demanding a ransom for the decryption of locked files. While ransomware attacks date back to the late 1980s, recent technological advancements have transformed it into a sophisticated, high-tech attack that leverages state-of-the-art encryption techniques [22] [23]. The prevalence of Ransomware as a Service (RaaS) has further increased the ease of launching ransomware attacks[6][1].

Ransomware attacks have become more anonymous[20] and challenging to trace, with attackers utilizing TOR routing protocols, Bitcoin for anonymous payments, and strong encryption techniques resistant to cracking attempts [24]. The evolving power and techniques in the hands of criminals necessitate the involvement of intelligence and advanced countermeasures in combating ransomware and other forms of malware.

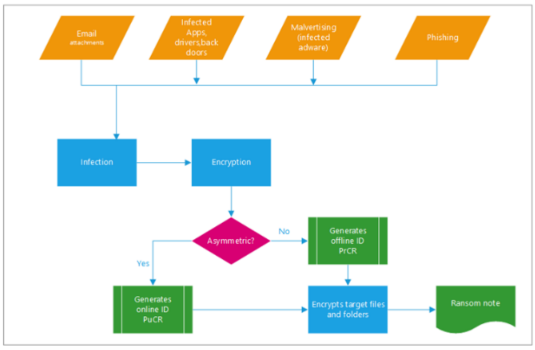

Email attachments
Infected Apps, drivers,back doors
Malvertising (infected adware)
Phishing
Infection
Encryption
Asymmetric?
No
Yes
Generates offline ID PrCR
Generates online ID PuCR
Encrypts target files and folders
Ransom note

Fig. 3. Crypto Ransomware infection activities flow

In light of these challenges, this article seeks to explore the potential of intelligence-based solutions in mitigating the adverse effects of ransomware attacks. By understanding the modus operandi and underlying technologies employed by ransomware[21][22], we can develop proactive measures to enhance cybersecurity and protect against these evolving cyber threats.

III. Related Works

To effectively address the challenges posed by ransomware, thorough research and understanding of the malware are crucial. Several studies have been conducted to demystify ransomware, its deployment, infection characteristics, and payment demands [29] [9]. Furthermore, researchers have explored the use of machine learning (ML) in detecting and analysing Crypto ransomware [30] [31].

Ransomware is a complex and dynamic threat, constantly evolving to overcome defences. ML, with its ability to detect patterns and mutations, has proven to be a valuable tool in combating malware, including Crypto ransomware [31]. Researchers have proposed various ML models and frameworks for Crypto ransomware detection [11]. Despite the progress made, a significant challenge remains: detecting unknown Crypto ransomware variants at the pre-encryption phase [9].

Many researchers have focused on dynamic and static analysis, or a combination of both, for ransomware detection. However, these approaches often fail to identify and detect emerging mutated variants [9]. In an effort to detect ransomware at the pre-encryption stage, a framework utilizing Frequency Centric Model, Datacentric detection, and Anomaly based detection was proposed [9]. However, determining the pre-encryption phase remains a challenge.

In contrast to reverse engineering ransomware binaries, some researchers have emphasized system behaviour analysis. They have developed neural network models that utilize system monitoring to create real datasets for ransomware detection [10] [32] [18].

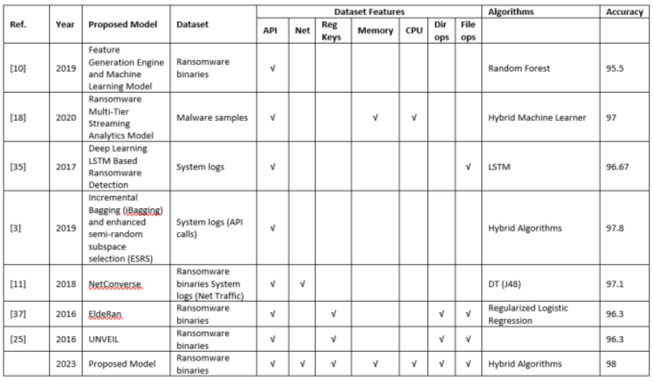
Signature-based detection, commonly used by antivirus systems, has been identified as a weak point in the fight against malware, including Crypto ransomware [32]. Behavioural analysis has been proposed as a key approach, with a hybrid framework combining static and dynamic analysis using API calls and invocations [32]. Deep learning, a subset of machine learning, has also been explored for enhanced detection of Crypto ransomware.

Multi-level ransomware detection framework leveraging big data platforms, natural language processing (NLP), machine learning, was proposed [34]. It utilized supervised ML algorithms and focused on the reverse engineering of known ransomware binaries. Another hybrid learner approach was utilized in a ransomware streaming analytics model, incorporating trait extraction, ancestor-family attribution, fusion, and a learning tier [18].

In addition to binary analysis, researchers have explored the classification of ransomware behaviour through API calls and invoking [35]. This breakthrough has overcome data limitations in the early stages of an attack. Furthermore, a pre-encryption detection algorithm (PEDA) was proposed to detect Crypto ransomware before any data encryption occurs, utilizing API calls to impede the encryption operation of the ransomware [4].

A. *Reviewed Models*

TABLE I. COMPARISON OF REVIEWED MODELS



Ref.	Year	Proposed Model	Dataset	Dataset Features							Algorithms	Accuracy
				API	Net	Reg Keys	Memory	CPU	Dir ops	File ops		
[10]	2019	Feature Generation Engine and Machine Learning Model	Ransomware binaries	√							Random Forest	95.5
[18]	2020	Ransomware Multi-Tier Streaming Analytics Model	Malware samples	√			√	√			Hybrid Machine Learner	97
[35]	2017	Deep Learning LSTM Based Ransomware Detection	System logs	√						√	LSTM	96.67
[3]	2019	Incremental Bagging (iBagging) and enhanced semi-random subspace selection (ESRS)	System logs (API calls)	√							Hybrid Algorithms	97.8
[11]	2018	NetConverse	Ransomware binaries System logs (Net Traffic)	√	√						DT (J48)	97.1
[37]	2016	EldeRan	Ransomware binaries	√		√			√	√	Regularized Logistic Regression	96.3
[25]	2016	UNVEIL	Ransomware binaries	√		√			√	√		96.3
	2023	Proposed Model	Ransomware binaries	√	√	√	√	√	√	√	Hybrid Algorithms	98

B. *Proposed Model*

The proposed model is created with the view of leveraging on the gaps observed from related works . The Model combines several behavioural patterns in its dataset features and is able to detect Crypto ransomware attacks in process. This model uses one of the robust and prominent ML classifiers called Random Forest [38][36]. The choice of this classifier over others is based on its advantageous attributes of resisting overfitting and the inversely proportionate attribute of decrease of variance with increase in number of trees, which as perceived should give very favourable accuracy score. However, this does not necessarily mean the model creation is limited to this ML algorithm. Because the proposed model is envisioned to utilize supervised learning, Decision Tress, Support Vector Machines, Naive Bayes and Logistic regression algorithms are considered in the classifier selection task and model creation.

C. *Classifiers*

The proposed system was developed based on five classification algorithms and the best performing algorithm was selected as our classifier for the model. The five classification algorithms were:

- Decision Tree Classifier
- Random Forest Classifier, and
- Logistic Regression.
- Naïve Bayes
- Support Vector

D. *Dataset*

A Dataset is a structured group of related facts or attributes of given entities. The proposed model made use of the Ransomwaredata2016 dataset [37] from the Kaggle repository.

This dataset contains the dynamic analysis of 582 samples of ransomware and 942 of good applications (good ware), i.e., 1524 samples in total. The dataset was retrieved and analysed with Cuckoo Sandbox at the end of February 2016. The dataset has data from 11 ransomware families with different sets of features.

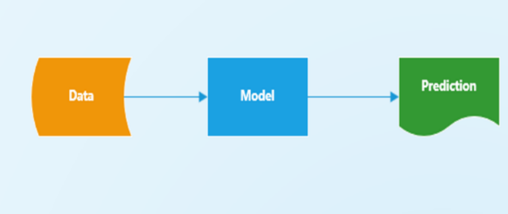


Fig. 4. Detection schema abstract

IV. METHODOLOGY

To develop a model for detecting Crypto ransomware attacks on Windows platforms, several steps are required. These include identifying the detection approach, extracting features, creating the model, and conducting training and testing. Currently, the most commonly used approach is based on file signatures, also known as a signature-based approach. However, this method is limited to detecting known Crypto ransomware variants as stated by [17]. It does not effectively address the detection of unknown variants or zero-day Crypto ransomware attacks, raising concerns about its efficacy in such cases.

This study aimed at creating a model that could be able to detect known and unknown Crypto ransomware variants effectively on Windows platforms. This was achievable by leveraging on the use of a dataset that had been created from dynamic analysis.

The detection schema had three main components: Pre-processing, Feature Selections and Classification. These three have been abstracted as shown in fig. 4 with all pre-processing activities being done under the data component, classification under Model and results being the prediction given by the model. The detailed description of the abstracted activities can be presented as shown in fig. 5. The process flow include data sourcing component, preprocessing and the actual testing of the dataset on the model for classification.

A. *Data Preprocessing*

There are several issues that may lead to reduced data quality[40]. These include missing values, too much data, too little data, inconsistency, and sparsity issues. As best practice, any ransomware model test plan begins with identification of problem then the gathering of a diverse and representative dataset consisting of both benign and ransomware samples.

This is followed by Data Preprocessing, which is simply cleaning and preprocessing the collected data to ensure uniformity and quality. This may involve converting files to a standardized format, removing duplicates, removing null values, and balancing the dataset to prevent bias towards any particular class.

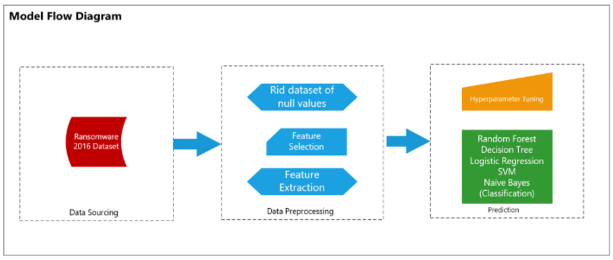
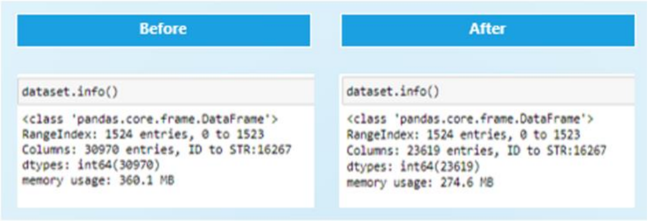


Fig. 5. Model flow diagram



```
dataset.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1524 entries, 0 to 1523
Columns: 30970 entries, ID to STR:16267
dtypes: int64(30970)
memory usage: 360.1 MB
```

```
dataset.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1524 entries, 0 to 1523
Columns: 23619 entries, ID to STR:16267
dtypes: int64(23619)
memory usage: 274.6 MB
```

Fig. 6. Data cleaning

B. Dataset Cleaning

In our model creation, the first step is to rid the dataset of columns and rows having null values. This reduces the dataset from 30970 to 23619, as shown in fig. 6.

C. Feature Selection

Feature selection techniques aim to identify the most relevant and discriminative features for ransomware detection, reducing the dimensionality of the data and improving the model's efficiency and performance [23]. Feature engineering involves creating new features or transforming existing ones to capture more meaningful information.

Using the ExtraTreesClassifier and SelectFromModel important features were selected from the dataset that could efficiently be subjected to training and testing. The result is 1723 features selected as important. This is shown Table II.

D. Feature Extraction

Feature engineering plays a vital role in developing a machine learning model that can distinguish between ransomware and benign samples. This process involves extracting relevant attributes from the dataset to enable effective learning. These attributes include file manipulation and modification characteristics, behavioral patterns related to resource effects, and specific traits unique to ransomware like file extensions and encryption signatures. Machine learning algorithms rely on meaningful features to learn, and feature extraction entails identifying pertinent attributes from ransomware samples and related data, such as file properties, system behavior, network traffic patterns, and API calls. By leveraging these extracted features, the model can better differentiate between ransomware and benign samples.

TABLE II. IMPORTAT FEATURES

IMPORTANT FEATURES
1. feature API:OpenSCManagerW (0.018848) 2. feature STR:43 (0.018082) 3. feature API:WriteConsoleW (0.017752)) 14. feature REG:OPENED:HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\User Agent\ (0.009237) 15. feature FILES_EXT:WRITTEN:addon (0.008871) 25. feature API:GetAsyncKeyState (0.006664) 26. feature STR:7282 (0.006217) 27. feature REG:OPENED:HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\bb1ed231d5eea329010213128f7f2ca476c4208a25ac323072962176b865b980.exe\ (0.005845) 1709. feature REG:READ:HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Ext\Stats\{1185823F-F22F-4027-80E5-4F68ACD5DE5E}\iexplore\ (0.000043) 1710. feature REG:OPENED:HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Shell Extensions\ (0.000043) \Cache\ (0.000042) 1719. feature FILES_EXT:OPENED:rgn (0.000042) 1720. feature API:OpenSCManagerA (0.000042) 1721. feature STR:13014 (0.000042) 1722. feature STR:5933 (0.000042) 1723. feature REG:WRITTEN:HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\WinDjView\ (0.000042)

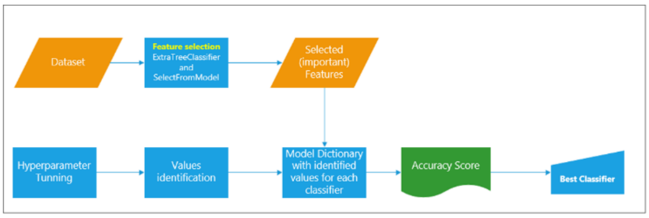


Fig. 7. Classifier selection process

Before inputting the train and test sets in the algorithm, there was need to compare a performance of a number of classifiers in order for us to come up with a best suited classifier. To identify the best options for tuning the classifiers, hyperparameter definition (identification of possible values) for each classifier, in conjunction with grid search with cross-validation was used. With the obtained hyperparameter values, a model dictionary was created to take the identified hyperparameter values, and using the corresponding accuracy score, the best performing algorithm was selected for use in the model

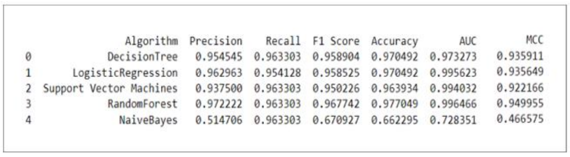
V. Results and Discussion

To evaluate the success of a research project, thorough testing and evaluation are necessary to determine its viability and impact. The choice of performance measures depends on the project's objectives and can include quantitative and qualitative indicators. When evaluating machine learning (ML) models, common performance measures such as accuracy, precision, recall, and the F1 score are used to assess effectiveness and generalization capabilities [24][25]. The confusion matrix provides insights into true positives, true negatives, false positives, and false negatives, while the receiver operating characteristic (ROC) curve visualizes performance trade-offs[26]. The area under the ROC curve (AUC) summarizes overall performance. Cross-entropy measures the discrepancy between predicted and actual outcomes, revealing the models' ability to capture underlying patterns[27]. Considering these metrics in the analysis enables a comprehensive evaluation, identification of strengths and weaknesses, and informed decision-making for optimal thresholds and future directions. This detailed analysis enhances transparency, reliability, and the ability to draw meaningful conclusions from the research findings.

A. Classifier Performance

The model creation was based on a best selected classifier from amongst five classifiers:

- Random Forest
- Decision Tree
- Logistic Regression
- Support Vector Machines
- Naïve Bayes



```
                Algorithm  Precision    Recall  F1 Score  Accuracy       AUC       MCC
0            DecisionTree   0.954545  0.963303  0.958904  0.970492  0.973273  0.935911
1      LogisticRegression   0.962963  0.954128  0.958525  0.970492  0.995623  0.935649
2  Support Vector Machines   0.937500  0.963303  0.950226  0.963934  0.994032  0.922166
3            RandomForest   0.972222  0.963303  0.967742  0.977049  0.996466  0.949955
4              NaiveBayes   0.514706  0.963303  0.670927  0.662295  0.728351  0.466575
```

Fig. 8. Metrics comparisons on Classifiers

As shown in fig. 8, the DecisionTree model achieved a precision of 0.9545, indicating that 95.45% of the detected ransomware samples were correctly classified. The recall score of 0.9633 demonstrates that the model identified 96.33% of the actual ransomware samples. The F1-score of 0.9589 represents a balance between precision and recall, indicating the model's overall performance.With an accuracy score of 0.9705, the DecisionTree model achieved an impressive classification accuracy of 97.05%. The area under the ROC curve (AUC) for DecisionTree was 0.9733, indicating a high level of discrimination between ransomware and benign files. The Matthews correlation coefficient (MCC) of 0.9359 indicates a strong correlation between the model's predictions and the actual outcomes.

The classification report for DecisionTree shows high performance in classifying both classes, with an accuracy of 97%. It provides detailed metrics such as precision, recall, and F1-score for each class, demonstrating the model's ability to differentiate between ransomware and benign files.

The LogisticRegression model achieved a precision of 0.9629, indicating a high level of accuracy in classifying ransomware samples. With a recall score of 0.9541, the model successfully identified a significant portion of the actual ransomware instances. The F1-score of 0.9585 indicates a good balance between precision and recall for the LogisticRegression model. Similar to the DecisionTree model, LogisticRegression achieved an accuracy score of 0.9705, indicating a high classification accuracy. The AUC value for LogisticRegression was 0.9956, demonstrating excellent discrimination capabilities. The MCC score of 0.9356 signifies a strong correlation between the model's predictions and the ground truth labels.

The classification report for LogisticRegression highlights accurate classification for both ransomware and benign files, with an overall accuracy of 97%.

The SVM model achieved a precision of 0.9375, indicating a high level of accuracy in classifying ransomware samples. With a recall score of 0.9633, the model successfully identified a significant portion of the actual ransomware instances. The F1-score of 0.9502 indicates a good balance between precision and recall for the SVM model. The SVM model achieved an accuracy score of 0.9639, indicating a high level of classification accuracy. The AUC value for SVM was 0.9940, indicating strong discrimination capabilities. MCC score of 0.9222 indicates a strong correlation between the model's predictions and the ground truth labels.

The classification report for SVM demonstrates good performance in accurately classifying both ransomware and benign files, with an overall accuracy of 96%.

The RandomForest model achieved a precision of 0.9722, indicating a high level of accuracy in classifying ransomware samples. With a recall score of 0.9633, the model successfully identified a significant portion of the actual ransomware instances. The F1-score of 0.9677 indicates a good balance between precision and recall for the RandomForest model.

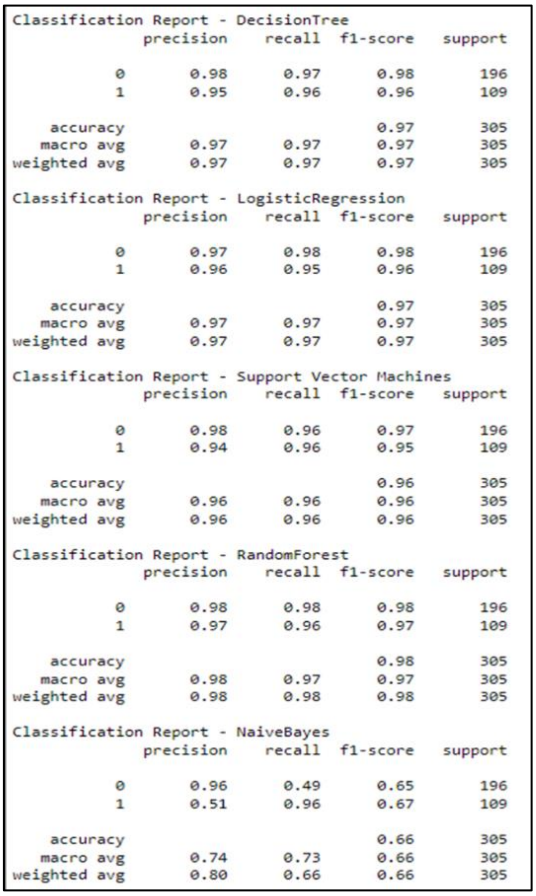
The RandomForest model achieved an accuracy score of 0.9770, indicating a high level of classification accuracy. The AUC value for RandomForest was 0.9965, demonstrating excellent discrimination capabilities. The MCC score of 0.9499 signifies a strong correlation between the model's predictions and the ground truth labels.

The classification report for RandomForest shows high accuracy in classifying both ransomware and benign files, with an overall accuracy of 98%.

The NaiveBayes model achieved a precision of 0.5147, indicating a lower level of accuracy in classifying ransomware samples compared to other models. With a recall score of 0.9633, the model successfully identified a significant portion of the actual ransomware instances. F1-score of 0.6709 indicates a balance between precision and recall, although it is relatively lower compared to other models. The NaiveBayes model achieved an accuracy score of 0.6623, indicating a moderate level of classification accuracy. The AUC value for NaiveBayes was 0.7284, suggesting limited discrimination capabilities. The MCC score of 0.4666 indicates a weaker correlation between the model's predictions and the ground truth labels compared to other models.

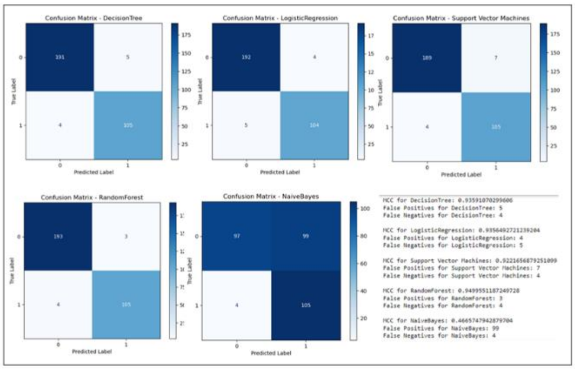
The classification report for NaiveBayes reveals imbalanced performance, with lower precision for class 0 (benign files) and higher precision for class 1 (ransomware).

In fig. 9 below, we give the Cross-Entropy Loss and Accuracy graphical representation. Both cross-entropy loss and accuracy are essential metrics for evaluating the performance of classification models. The cross-entropy loss helps in training the model by providing a measure of the discrepancy between predicted probabilities and true labels. Accuracy provides a straightforward measure of the model's ability to correctly classify instances and is useful for overall model evaluation.



```
Classification Report - DecisionTree
              precision    recall  f1-score   support

           0       0.98      0.97      0.98       196
           1       0.95      0.96      0.96       109

    accuracy                           0.97       305
   macro avg       0.97      0.97      0.97       305
weighted avg       0.97      0.97      0.97       305

Classification Report - LogisticRegression
              precision    recall  f1-score   support

           0       0.97      0.98      0.98       196
           1       0.96      0.95      0.96       109

    accuracy                           0.97       305
   macro avg       0.97      0.97      0.97       305
weighted avg       0.97      0.97      0.97       305

Classification Report - Support Vector Machines
              precision    recall  f1-score   support

           0       0.98      0.96      0.97       196
           1       0.94      0.96      0.95       109

    accuracy                           0.96       305
   macro avg       0.96      0.96      0.96       305
weighted avg       0.96      0.96      0.96       305

Classification Report - RandomForest
              precision    recall  f1-score   support

           0       0.98      0.98      0.98       196
           1       0.97      0.96      0.97       109

    accuracy                           0.98       305
   macro avg       0.98      0.97      0.97       305
weighted avg       0.98      0.98      0.98       305

Classification Report - NaiveBayes
              precision    recall  f1-score   support

           0       0.96      0.49      0.65       196
           1       0.51      0.96      0.67       109

    accuracy                           0.66       305
   macro avg       0.74      0.73      0.66       305
weighted avg       0.80      0.66      0.66       305
```

Fig. 9. Classification report for all five classifiers


Confusion Matrix - DecisionTree
191
5
4
105
Confusion Matrix - LogisticRegression
192
4
5
104
Confusion Matrix - Support Vector Machines
189
7
4
105
Confusion Matrix - RandomForest
193
3
4
105
Confusion Matrix - NaiveBayes
97
99
4
105
True Label
Predicted Label
0
1

```
MCC for DecisionTree: 0.93591070299606
False Positives for DecisionTree: 5
False Negatives for DecisionTree: 4

MCC for LogisticRegression: 0.9356492721239204
False Positives for LogisticRegression: 4
False Negatives for LogisticRegression: 5

MCC for Support Vector Machines: 0.9221656879251099
False Positives for Support Vector Machines: 7
False Negatives for Support Vector Machines: 4

MCC for RandomForest: 0.9499551187249728
False Positives for RandomForest: 3
False Negatives for RandomForest: 4

MCC for NaiveBayes: 0.4665747942879704
False Positives for NaiveBayes: 99
False Negatives for NaiveBayes: 4
```

Fig. 10. Confusion Matrix for all Classifiers

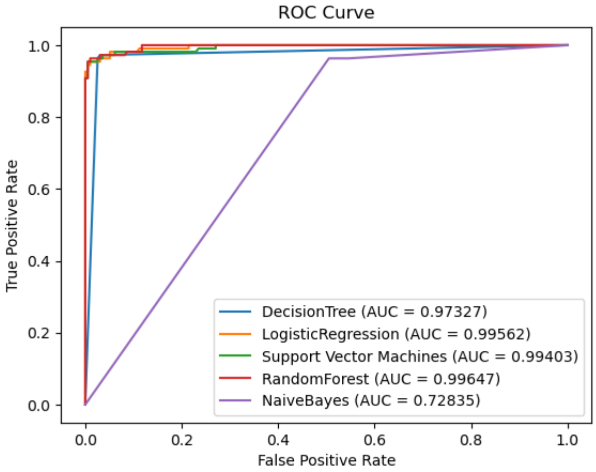


Fig. 11.

The results of the classifications for each of the classifiers were further represented as shown in the combined confusion matrix in Fig 10. As shown in Fig. 11, all the classifiers are above the random classifier line which indicates they are better than random performance, however RandomForest is unarguably closer to the top left corner denoting that it has perfect discrimination and is highly desirable.

B. *Comparison with existing research*

The model demonstrates exceptional performance, with an impressive accuracy score of 98%. This surpasses the achievements of similar studies conducted by (Mkandawire and Zimba, 2023) using Logistic Regression (97.70% accuracy with the same dataset), (Sgandurra et al., 2016) using EldeRan (96.3% accuracy), and (Zuhair, Selamat and Krejcar, 2020), among others with (97% accuracy).

Our model boasts of improved performance and score owing to techniques used. These included:

1. Ridding the data frame of null values led to a reduction in the number of features that would later be subjected to the ExtraTreesClassier tool.
2. Non-use of standardization techniques such as min-max scaling. Such techniques are not applicable to binary features. In our model development, standardization had no effect and actually reduced the accuracy score.
3. Use of ExtraTreesClassifier and SelectFromModel for feature extraction works efficiently by helping obtain a reduced feature set containing the most important features.

VI. Conclusion

Crypto ransomware is no longer a growing menace, but a deep rooted one that requires the utilization of available favorable technologies to control and or uproot it. With the world of technology inclining towards ML, there is growing research in the use of ML to find effective ways of fighting Crypto ransomware. Studies have shown that it is easier to detect known ransomware, than it is for new variants. These emerging variants are now mostly characterized with mutated traits which make it difficult for Antivirus systems to detect them.

This study presents a viably prospective solution to help mitigate the spread of Crypto ransomware attacks by detecting when these attacks are in process, during the many inconspicuous activities that take place between target acquisition and encryption of files, commonly referred to as the pre-encryption phase. Having identified the gaps in the ransomware detection research works considered, this study identifies an inclusion of a wider dataset features that can be key in Crypto ransomware detection, covering a wide range of ransomware behavioral patterns and the utilization of appropriate techniques in model tuning.

Acknowledgment

I wish to extend sincere gratitude to Mulungushi University for sponsoring this study. Special thanks to my Supervisors Dr. Aaron Zimba (Ph. D.) and Dr. Muwanei Sinyinda (Ph. D.), for above everything that can be listed, the motivation, critique and help they gave throughout the course of this research work. I would also like to recognize the contribution made by Dr. Mwenge Mulenga (Ph. D.) for all his valuable insights in ML and Datamining shared throughout the research work.

References

[1] E. Kolodenker, W. Koch, G. Stringhini, and M. Egele, “PayBreak : Defense against cryptographic ransomware,” *ASIA CCS 2017 - Proc. 2017 ACM Asia Conf. Comput. Commun. Secur.*, pp. 599–611, 2017, doi: 10.1145/3052973.3053035.

[2] U. Urooj, B. A. S. Al-Rimy, A. Zainal, F. A. Ghaleb, and M. A. Rassam, “Ransomware Detection Using the Dynamic Analysis and Machine Learning: A Survey and Research Directions,” *Appl. Sci.*, vol. 12, no. 1, 2022, doi: 10.3390/app12010172.

[3] Johnson J, “Number of ransomware attacks per year 2020 | Statista,” *Statista*, 2021. https://www.statista.com/statistics/494947/ransomware-attacks-per-year-worldwide/

[4] Sana Aurangzeb, Muhammad Arshad Islam, Muhammad Azhar Iqbal, and Muhammad Aleem, “Ransomware: A Survey and Trends,” *J. Inf. Assur. Secur.*, vol. 6, no. 2, pp. 48–58, 2017.

[5] B. A. S. Al-rimy, M. A. Maarof, and S. Z. M. Shaid, “Crypto-ransomware early detection model using novel incremental bagging with enhanced semi-random subspace selection,” *Futur. Gener. Comput. Syst.*, vol. 101, pp. 476–491, 2019, doi: 10.1016/j.future.2019.06.005.

[6] S. H. Kok, A. Azween, and N. Z. Jhanjhi, “Evaluation metric for crypto-ransomware detection using machine learning,” *J. Inf. Secur. Appl.*, vol. 55, p. 102646, 2020, doi: 10.1016/j.jisa.2020.102646.

[7] P. O’Kane, S. Sezer, and D. Carlin, “Evolution of ransomware,” *IET Networks*, vol. 7, no. 5, pp. 321–327, 2018, doi: 10.1049/iet-net.2017.0207.

[8] Y. Feng, C. Liu, and B. Liu, “Poster ： A New Approach to Detecting Ransomware with Deception,” *38th IEEE Symp. Secur. Priv.*, pp. 3–4, 2017.

[9] B. A. S. Al-rimy, M. A. Maarof, and S. Z. M. Shaid, “A 0-day aware crypto-ransomware early behavioral detection framework,” *Lect. Notes Data Eng. Commun. Technol.*, vol. 5, pp. 758–766, 2018, doi: 10.1007/978-3-319-59427-9_78.

[10] J. Hernandez-Castro, E. Cartwright, and A. Stepanova, “Economic Analysis of Ransomware,” *SSRN Electron. J.*, pp. 1–14, 2017, doi: 10.2139/ssrn.2937641.

[11] I. Yaqoob *et al.*, “The rise of ransomware and emerging security challenges in the Internet of Things,” *Comput. Networks*, vol. 129, pp. 444–458, 2017, doi: 10.1016/j.comnet.2017.09.003.

[12] P. Mahendru, “The State of Ransomware in Retail 2022,” no. May, 2022.

[13] S. Poudyal, K. P. Subedi, and D. Dasgupta, “A Framework for Analyzing Ransomware using Machine Learning,” *Proc. 2018 IEEE Symp. Ser. Comput. Intell. SSCI 2018*, pp. 1692–1699, 2019, doi: 10.1109/SSCI.2018.8628743.

[14] O. M. K. Alhawi, J. Baldwin, and A. Dehghantanha, “Leveraging machine learning techniques for windows ransomware network traffic detection,” *Adv. Inf. Secur.*, vol. 70, pp. 93–106, 2018, doi: 10.1007/978-3-319-73951-9_5.

[15] I. Bello *et al.*, “Detecting ransomware attacks using intelligent algorithms: recent development and next direction from deep learning and big data perspectives,” *J. Ambient Intell. Humaniz. Comput.*, vol. 12, no. 9, pp. 8699–8717, 2021, doi: 10.1007/s12652-020-02630-7.

[16] Statista Research Department, “Number of ransomware attacks per year 2016-H1 2022,” *Statista*, 2022. https://www.statista.com/statistics/700965/leading-cause-of-ransomware-infection/

[17] N. Hampton, Z. Baig, and S. Zeadally, “Ransomware behavioural analysis on windows platforms,” *J. Inf. Secur. Appl.*, vol. 40, pp. 44–51, 2018, doi: 10.1016/j.jisa.2018.02.008.

[18] N. Rani and S. V. Dhavale, “Leveraging Machine Learning for Ransomware Detection,” 2022.

[19] A. Zimba, Z. Wang, and H. Chen, “Reasoning crypto ransomware infection vectors with Bayesian networks,” *2017 IEEE Int. Conf. Intell. Secur. Informatics Secur. Big Data, ISI 2017*, pp. 149–151, 2017, doi: 10.1109/ISI.2017.8004894.

[20] A. Zimba, L. Simukonda, and M. Chishimba, “Demystifying Ransomware Attacks: Reverse Engineering and Dynamic Malware Analysis of WannaCry for Network and Information Security,” *Zambia ICT J.*, vol. 1, no. 1, pp. 35–40, 2017, doi: 10.33260/zictjournal.v1i1.19.

[21] A. Kapoor, A. Gupta, R. Gupta, S. Tanwar, G. Sharma, and I. E. Davidson, “Ransomware detection, avoidance, and mitigation scheme: A review and future directions,” *Sustain.*, vol. 14, no. 1, pp. 1–24, 2022, doi: 10.3390/su14010008.

[22] M. A. Sotelo Monge, J. M. Vidal, and L. J. García Villalba, “A novel self-organizing network solution towards crypto-ransomware mitigation,” *ACM Int. Conf. Proceeding Ser.*, 2018, doi: 10.1145/3230833.3233249.

[23] Y. Mkandawire and A. Zimba, “A Supervised Machine Learning Ransomware Host-Based Detection Framework,” vol. 7, no. 1, pp. 52–56, 2023.

[24] C. Chebbi, *Mastering Machine Learning for Penetration Testing*, vol. 53, no. 9. 2018.

[25] S. Bhatore, L. Mohan, and Y. R. Reddy, “Machine learning techniques for credit risk evaluation: a systematic literature review,” *J. Bank. Financ. Technol.*, vol. 4, no. 1, pp. 111–138, 2020, doi: 10.1007/s42786-020-00020-3.

[26] D. Sgandurra, L. Muñoz-González, R. Mohsen, and E. C. Lupu, “Automated Dynamic Analysis of Ransomware: Benefits, Limitations and use for Detection,” 2016.

[27] R. Sujatha, S. L. Aarthy, and R. Vettriselvan, “Integrating Deep Learning Algorithms to Overcome Challenges in Big Data Analytics,” *Integrating Deep Learning Algorithms to Overcome Challenges in Big Data Analytics*. 2021. doi: 10.1201/9781003038450.