A MACHINE LEARNING MODEL FOR DETECTING AND CLASSIFICATION OF RANSOMWARE.

Authors: Asogwa D.C[1], Orah R.O[2], AnusiubaO. I[3], Mbonu C.E[4]
Department of Computer Science, Faculty of Physical Sciences, NnamdiAzikiwe University, Awka. Anambra state Nigeria [1, 3, 4]. College of Information and Technology, Salem University, Lokoja, Kogi state, Nigeria[2].
dc.asogwa@unizik.edu.ng[1], orahseun@gmail.com[2],oi.anusiuba@unizk.edu.ng[3],ce.mbonu@unizik.edu.ng[4]

DOI: 10.26821/IJSHRE.9.8.2021.9806

ABSTRACT

The dynamic concept of technology has caused an unprecedented technological and socio-economic development in everyday human activities. The fact is that there is an increasing number of digital attacks and digital kidnapping, purporting to be ransomware as a continuing threat. This has resulted in the battle between the development and detection of new techniques. Detection and mitigation systems have been developed and are in wide-scale use. However, their reactive nature has resulted in a continuing evolution and updating process. This is largely because detection mechanisms can often be circumvented by introducing changes in the malicious code and its behaviour. In this paper, classification techniques were used to develop a machine learning model for the detection and classification of ransomware. This will also increase the accuracy of the detection and classification of ransomware. Supervised machine learning algorithms were trained for building the model and the test set used to perform the model evaluation using confusion matrix. This ensured a systematic comparison of each algorithm. The supervised algorithms used are naive Bayes and decision tree (J48). This resulted to an accuracy of 83.40% for Naïve Bayes and 97.60% for Decision Tree (J48).The Research also determine sensitivity and specificity.

Keywords: Digital Files, Ransomware, filtering, digital kidnaping attacks, machine learning model

1. INTRODUCTION

It has been revealed that in this modern world, the use of digital technology devices is gaining ground every day, the negative outcome (threats) of these digital technology devices are also growing rapidly. There are a lot of malicious programs, such as virus, worm, Trojan, backdoor, or spyware, which can seriously damage or harm digital technology systems. Among all the current malicious software, ransomware appears to be one of the most alarming. Ransomware is a form of malware that encrypts a victim's files. It is a type of malware virus that restricts users' access to the system and their data, by encrypting user files or locking the system and then request ransom payment to give back the access to the users. Due to advance in technology over the period of years, ransomware attacks have seen a significant growth in numbers and it has caused a lot of damages to a digital file or personal computer (PC). Cyber Criminals are getting more new tactics as well as innovative, and the damage is only getting worse.

According to a study by Datto, a leading cybersecurity company [1], ransomware is responsible for more than US $77 billion extortion yearly. Even the healthcare and financial service industries are the top targets of this attack. Over 50% of the participants in the study believed their business was not ready to handle ransomware threat.

CryptoLocker'sransomware has infected approximately 280 thousand computers system worldwide, including an entire police department which had to pay a ransom to decrypt their files or documents, innovated by Kharraz A., Robertson W., Balzarotti D., Bilge L., and Kirda E., [2]. In 2017, NotPetya and Wannacryransomware were wakeup calls to businesses all around the world. The Hollywood Presbyterian Medical Center, in February 2016, paid a ransom amount of 40 Bitcoins valued $17,000 at the time after being hit by a ransomware attack that crashed the hospital's entire network, Richard Winton, [3].

Research shows that in May 2016, the University of Calgary paid US $16,129 after ransomware handicapped multiple systems UToday, [4].

The first ransomware ever used was PC CYBORG/AIDS. It was delivered using a floppy disk, and it mainly counts for the number of times the system reboots. When system reboot count reaches 90, it hides directories and encrypts all the file names in the system root directory, proposed by Shinde R., Van Der Veeken P., Van Schooten S., and Van Den Berg J., [5]. Until a few years ago, ransomware incidents were not significant.

However, with the evolution of robust encryption techniques, ransomware started making headlines as the most notable malware, and as mentioned above, ransomware infections have cost users a considerable amount of time and money over the past several years.

There are two main types of ransomware currently available: locker-ransomware and crypto-ransomware. Locker-ransomware locks the computer system to prevent the user from using it. Crypto-ransomware encrypts the user's files or document to make them inaccessible to victims. Most often crypto-ransomware does not encrypt the entire hard-disk but searches for specific extensions only. The user is threatened to pay a ransom by holding hostage her data or system. Users can regain access to their files only through anonymous payment mechanisms, such as crypto currencies.

The amount of money requested by ransomware is different, but it is usually between 400 to 800, which is paid through Bitcoin proposed by Cabaj K., et al., [6]. If the victim does not accept to pay the ransom, he cannot use its files. Hence, crypto ransomwares are more dangerous than locker-ransomwares. Therefore, ransomwareare most often consideredas one of the most dangerous types of malwares. There is also a combination of locker/crypto ransomware where a user is blocked from using their computer while their data is being encrypted

In fact, the aim of this kind of malware known as ransomware is to earn money, which gives more motivation to the attackers. Consequently, effective and efficient methods or techniques are required for ransomware detection and classification. Although many models have been developed to detect and classify the ransomwares, but many of these techniques have failed to detect new ransomwares. The aim of this research is to develop a machine learning model for detection and classification of ransomwares. The process is a binary classification which focuses on (ransomwares and benign) using data mining techniques and classification algorithms. In this research, after extracting the process model using the WEKA data mining tool, features of this model were extracted, then using these features and classification algorithms, ransomwares can be identified.

Theobjectives includeapplying machine learning algorithm to develop a model that will detect and classify ransomware attacks in digital files using Naïve Bayes algorithm and Decision Tree.

Many users, institutions, and organizations have been exposed to ransomware threat, resulting in major financial and reputation loss. So, it is important to design a model that can detect and prevent ransomware efficiently with the minimum false positive at lower costs.

The paper is organized as follows: section II discusses the related works on ransomware, section III analyzes the system problem, giving details of the methodology, dataset description, and feature set description and the experimental setup. Section IV presents the system implementation, evaluation and discussion of the results. Section V presents the conclusion of the report, recommendation and point out areas for future works

2. RELATED WORKS

In 2015, Kharraz*el al*, [2] studied ransomware attacks that occurred in the wild from 2006 to 2014. The study explored 15 different ransomware families and showed that almost 94% ransomware samples implement simple locking or encrypting techniques. The authors suggested that by closely monitoring file system activity and the types of I/O request packets to the file system, it is possible to detect ransomware attacks. They also observed that Bitcoin addresses used to collect ransom payment from victims share similar transaction records, such as a small number of transactions, small Bitcoin amounts, short activity period, etc. However, despite proposing possible strategies for ransomware detection, no concrete experimental evaluation was conducted by the authors.

In the follow-up work presented by Kharraz*et al*., [2] a ransomware detection system called UNVEIL was proposed. UNVEIL looks at the file system layer to spot the typical ransomware behavior. It uses text analysis techniques to detect ransomware threatening notes and continuously takes screenshots of the desktop to check for screen lockers. It also uses statistical analysis based on memory usage, processor usage, and disk I/O rates to detect abnormal behavior for ransomware variants. The experimental evaluation yielded 96.3% accuracy in detecting ransomware. Despite achieving relatively high accuracy, the model does not have early detection capability for ransomware attacks nor does it provide any backup mechanism. Also, the proposed system is inherently reactive and ineffective for newer ransomware samples.

On the other hand, ShieldFS, a competitor to UNVEIL developed by Continella*et al*. in 2016, [18] is a self-healing ransomware-aware detection system with the additional capability of allowing the system to roll back malicious changes. It internally monitors low-level file system activities by computing the entropy of write operations, and the frequency of read, write, and folder listing operations. It also searches the memory regions of any process considered as potentially malicious, by looking specifically for block cipher key schedules. The system combines both automatic detection and transparent file recovery in a ready to use driver. However, this methodology also has some limitations as new variants of ransomware tend to encrypt or delete the Windows shadow copy of the file system, making the chances of file recovery almost zero. Additionally, the system is more focused on file operation related features only. The memory scanning aspect is time consuming and is plagued by the fact that there are rare chances to find a key in memory region.

CryptoDrop was an early warning detection system to alert users during suspicious file activities. The system mainly focused on monitoring user data for changes. The authors divided ransomware into three major classes: class A, class B, and class C based on how they encrypt the user files. They used similarity functions to measure the dissimilarity between the original and the encrypted contents of each file. CryptoDrop was unable to determine the purpose of the changes in its audit. For example, it was not able to differentiate between the user-triggered encryption and ransomware triggered-encryption.

Sgandurraet al in 2016 presented a machine learning approach called EldeRan, [19], for analyzing and detecting ransomware. In the first phase, EldeRan monitors a set of activities performed by applications and checks for attributes of ransomware. In the second phase, features like API calls, dropped files, registry keys, and directory enumerations are fed to a regularized machine learning model to learn patterns to differentiate between ransomware and benign applications. The experimental evaluation was based on a dataset involving 582 ransomware from 11 different families. An accuracy of 96.3% was obtained using dynamic analysis with a limited number of features. EldeRan was not able to extract the features when ransomware was silent for some time. Additionally, most of the features used in this system were binary. The authors focused only on the absence or presence of some of the features like registry key operations, mutex, etc. However, in the new variants of malware, the absence of these particular operations makes the detection model ineffective. For example, a registry key operation used in one variant of ransomware might not be used by other variants or new versions of ransomware.

Chen *et al*., [20], proposed an approach for ransomware detection based on dynamic API calls flow graph by monitoring API call sequences of malware binaries and converting them to a set of features. They used different data mining algorithms including random forest, SVM, Naive byes and logistic regression. The logistic regression achieved the highest accuracy of 98.2% with the lowest false positive rate of 1.2%. However, the focus was only on a single feature to detect ransomware and the evaluation was based on a dataset consisting of only 168 ransomware samples.

Lanzi*et al*. [21] collected a large number of system calls from regular users on actual inputs and studied the diversity of system and API calls. They observed that the interactions of benign programs with the operating system are different from those of malicious programs.

Kumar *et al*., [2], leveraged the dominance of API invocations to build a multi-layer perceptron (MLP), neural network model. Experimental evaluation of the proposed model on a dataset consisting of 7 different ransomware families yielded an accuracy of 98%.

Poudyal*et al*. [22] developed a reverse engineering framework for malware detection. The authors conducted a multi-level analysis of assembly codes, libraries and function calls, and applied different supervised machine learning techniques, including Bayesian Network, Random Forest, Smo and J48. The experimental evaluation yielded a detection accuracy of ransomware samples ranging from 76% to 97% based on the machine learning techniques used.

Recently, several works have been published on ransomware detection for mobile phones and the Internet of Things (IoT) as well. Karimi and Moattar [23] presented an approach that transforms a sequence of executables into a grey scale image. Then, they used Linear Discriminant Analysis (LDA) statistical method to separate two or more classes with dimension reduction functionality to improve the performance of the model. The evaluation of the proposed model was conducted through two different experiments. The first experiment was conducted using a dataset consisting of 140 ransomware samples from two well-known families and 20 benign samples, yielding 97% accuracy. In the second experiment, the model achieved an accuracy of 97.3% with a dataset consisting of 230 ransomware samples from Locker and Koler families and 30 benign samples.

Andronio*et al*, [24] studied mobile ransomware families on Android devices, and introduced an approach, named HelDroid, to discriminate known and unknown ransomware samples from benign applications. HellDroid tracks and detects ransomware behavior at the application layer and uses Natural Language Processing (NLP) to recognize threatening phrases. The evaluation of the system achieved accuracy over 97% with a dataset consisting of 650 ransomware and about 81,000 benign samples. However, detection of threatening phases is not much useful as by the time the user gets a ransom note on the screen the data is already encrypted.

This paper provides the background knowledge on ransomware, and then summarized and discussed the related work on ransomware detection. Most of the research works discuss feature extraction techniques and machine learning models that could be applied to distinguish benign and ransomware behaviors correctly.

It is clear from the reviewed research that classification model using static analysis is not enough to classify the ransomware effectively. Furthermore, behavioral based ransomware detection system is more effective than static based system for the detection of new ransomware. From the above literature analysis, one can also note that most of the work focuses on a limited number of features like API calls monitoring and file operations. As a result, ransomware which do not use default Windows APIs are hard to detect with the existing models. Also, existing models are incapable of distinguishingransomware encryption from user encryption.

While registry-key operations and file entropy were considered in one way or another in the existing literature, there has not been any systematic focus on how to utilize them in combination to detect ransomware. This work introduces a machine learning approach for development of ransomware detection and classification with two new sets of features: grouped registry key operations and combined file-signature and file-entropy. The benefit of using the aforementioned features is three-fold: improved accuracy, improved new ransomware detection rate, and helping identify user triggered and ransomware triggered encryption.

3. MATERIALS AND METHOD

The methods were based on the sample of Ransomware and Benign concerning the contents of the Ransomware threat. The following approaches were used for achieving this report: with respect to their classes. The system was built with the available data set collected via online resources with other related literature review such as journal or articles.

The analysis of the system methodology is based on the concept of the following:

3.1 Machine learning Approach

1) Collection of the sample data (Ransomware, Benign/historical data)

2) Pre-processing (the data were provided with two labels, Ransomware, and Benign), since it is a supervised learning approach, and also a binary classification

3) Feature extraction with Weka library (to convert the Ransomware, Benign into binary class values) using string to word vector

4) Resample the dataset by applying training set and testing set during system development analysis using Weka tools.

5) Apply machine learning model (Naïve Bayes and Decision Tree) to the trained dataset for the implementation of the new system.

6)Deploy the model for decision making to the stakeholder

The model was developed usingWeka plugin and java Netbeans GUI to implement the system with the entire requirement stated above.The following algorithms were used to perform the classification model and structured data analytics.

3.2 Naïve Bayes Classifier

Naïve Bayes is a supervised algorithm which is popular for text classification algorithm due to high speed and good performance. Based on the training set provided, it outputs where:

- ✓ p (Ck | x) is posterior.
- ✓ p (Ck) is prior.
- ✓ p (x | Ck) is likelihood.
- ✓ p (x) is evidence.

After having information from the dataset, the dataset was being calculated by the probability equation to make the detection and classify distinguishable. The calculations would be for the threat and benign data by forming the data into values 1's and 0's, while the 1 is for the threat data and the 0 is for the benign ones. Seventeen types of threat data have been discovered during this work, they were all different types of malware; all of them were worm type. The algorithm has two counters (i and j) first one for threat and the other for benign data, when the system detects abnormal behavior the counter (i) will increment its value by 1; the same goes for j counter but for the benign data when normal behavior detected it will increment its value by 1. The equation will calculate the values of i and j and find how much i and j incremented and then it will classify the data and show the results of threat and benign.

3.3 Decision Tree Algorithm

There may be a situation where instance has no value for a missing attribute or may have some unknown value. In this case, the missing value may be replaced by the value that occurs most common in training instances for which it is being tested successfully. In the algorithm below, each and every possible value taken by the attribute having missing value can be calculated on the number of times that an instance can be seen in the training instances at a node. The algorithm of the j48 is as follows:

3.3.1Algorithm (J48Tree)

```
INPUT
DataSet/Training data
OUTPUT
Tree//Decision tree
BUILD(*DataSet)
{
Tree = φ;
Tree = Create node as root and label
with splitting attribute predicate
and labels are assign;

for each are do
DataSet = Database created by applying splitting to
Dataset;
If stopping point reached to this path, then
Tree = create leaf node and label with appropriate,
class
Else
Tree= BUILD(DataSet);
Tree= add Tree to arc
}
```

4 System Design

The methods used to achieve this work are as follows:

Data collection

Data preprocessing

Feature extraction

Training set and Test set

Building the model

Based on the above, supervised learning will be used for training of the algorithm with labeled data as to which class it belongs. Using the labeled data, the algorithm learns the relationship between the feature sets and the output, and hence it then classifies the unlabeled data from the learned relationship as shown in figure 1. Here is conceptual framework of the model

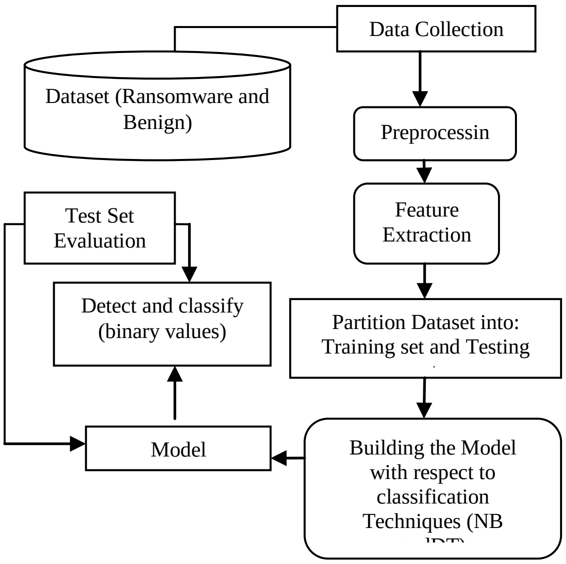
Data Collection
Dataset (Ransomware and Benign)
Preprocessin
Feature Extraction
Test Set Evaluation
Detect and classify (binary values)
Partition Dataset into: Training set and Testing
Model
Building the Model with respect to classification Techniques (NB

Figure 1 Steps for Ransomware classification

Pre-Processing

In this step complete geometric correction and filtering is done. The preprocessing uses the output of the classifier to take the required action to improve the performance

Dataset Description

The dataset used in this paper is available on internet. Virus Share (virusshare.com) was used to access a ransomware repository and goodware were acquired from Portable Apps (portableapps.com). Kali Linux was used for analyzing of the malware samples. In total, 400 executable (200 malwares and 200 goodware) were used. First, the executables were put through an online intelligence platform (virus-total.com). The use of virtual machine was a necessity to be able to see the features of the malware and goodware properly without the risk of damaging or condemning the host machine being used for the analysis. The working flow for Ransomware detection and classification is shown in figure 2.

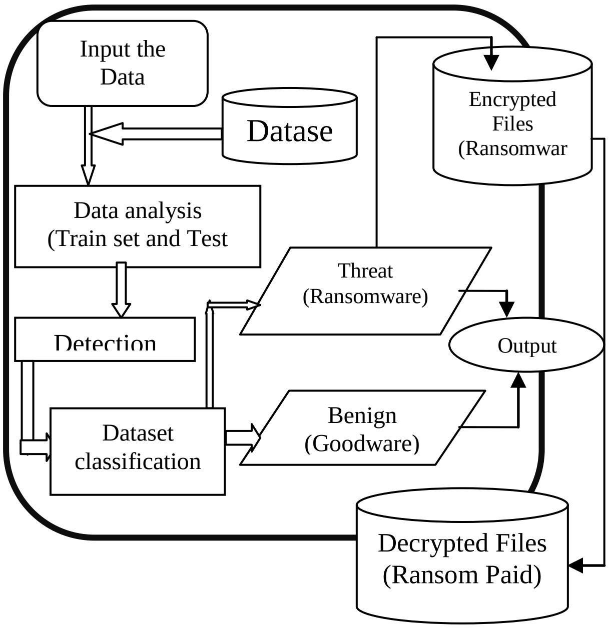

Figure 2 working flow of RansomwareProcesses

Experimental Set Up

All the experiments that were carried out in this section are computed using open source tool Weka 3.8.0[26] and java programming language with Netbeans IDE under the OS Windows for implementation of the machine learning model and processor with 4GB of main memory. Weka is a collection of a machine learning algorithm for data mining tasks. These algorithms may be applied directly using the default algorithm in the tool itself or we can call the algorithm using java code.

Selection of Training Data

In this step the particular attributes were selected which best describes the pattern for detecting either the benign or ransomware.

Classification of Outputs

The output of the expected result is classified as to different categories accordingly namely benign or ransomware.

5 IMPLEMENTATION

This was achieved with the two models namely Decision Tree and Naïve Bayes. These algorithms were used to train the data collected and the model was built by calling the java code without using the default algorithm. These Ransomware datasets were pre-processed and features were extracted before applying a classification algorithm on it. The classification method used was also based on the Decision Trees and Naïve Bayesalgorithms which were able to capture all the required training sets and used for the prediction/detection and classification.

The system was implemented with the set of seventeen (17) feature set or attributes to distinguish their performance when those factors were structured into the Weka plugin and java Netbeans to classify those sample dataset which were implemented as an information table. Thesample collected via online resource were seventeen attributes and (62485) instances used to perform the analysis and also used to build the model for predicting a promising result. Here the numeric values from a given sample were transformed into an excel format with an extension of csv and arff for machine readable task

Model Evalaution

The Ransomware and Benign analysis was done on two folds, the training set that was used to build the mode l(Naïve Bayes) and then the test set for detection and classification of the results with an unknown class labels to predict a new class label with their respective binary class values. Below is the classification results on Ransomware and Benign with respect to their parameters/attributes selected:

Correctly Classified Instances 1668 83.4 %
Incorrectly Classified Instances 332 16.6 %
Kappa statistic 0.6682
K&B Relative Info Score 125878.1554 %
K&B Information Score 1258.7637 bits 0.6294 bits/instance
Class complexity | order 0 1999.9724 bits 1 bits/instance
Class complexity | scheme 3679.1241 bits 1.8396 bits/instance
Complexity improvement (Sf) -1679.1517 bits -0.8396 bits/instance
Mean absolute error 0.1902
Root mean squared error 0.365
Relative absolute error 38.0397 %
Root relative squared error 73.0027 %
Total Number of Instances 2000

Table 1.0: Results and Analysis with Naïve Bayes

Class	***Precision***	***Recall***	***F-Measure***	***ROC Area***
Benign	*0.794*	*0.901*	*0.844*	*0.920*
ransomware	*0.886*	*0.768*	*0.823*	*0.892*

The results and analysis with Naïve Bayes can be seen in Table 1.0.

The analysisof Ransomware and Benign which was done on two folds, the training set that was used to build the model (Decision Tree (J48)) with train set and then the test set for detection and classification of the results with an unknown class labels to predict a new class label with their respective binary class values. Below are the classification results on Ransomware and Benign with respected to their parameters/attributes selected:

Correctly Classified Instances 1952 97.6 %
Incorrectly Classified Instances 48 2.4 %
Kappa statistic 0.952
K&B Relative Info Score 186264.5626 %

K&B Information Score 1862.6191 bits

Class	*Precision*	*Recall*	*F-Measure*	*ROC Area*
Benign	*0.986*	*0.966*	*0.976*	*0.975*
Ransomware	*0.967*	*0.986*	*0.976*	*0.975*

0.9313 bits/instance

Class complexity | order 0 1999.9724 bits 1 bits/instance

Class complexity | scheme 10984.7838 bits 5.4924 bits/instance

Complexity improvement (Sf) -8984.8114 bits -4.4924 bits/instance

Mean absolute error 0.038

Root mean squared error 0.1522

Relative absolute error 7.6071 %

Root relative squared error 30.4451 %

Total Number of Instances 2000

Table 2.0 Results and Analysis with Decision Tree

Algorithm	**Accuracy %**	**Train Set**	**Test Set**	**Binary Class**
Naïve Bayes	83.40%	8000	2000	Ransomware & Benign
Decision Tree	97.60%	8000	2000	Ransomware & Benign

Algorithm Comparison

Table 3.0 Detail Performance Evaluation by class

Table 3.0 showsthe detailed performance evaluation by class of the two supervised machine learning (Naïve Bayes and Decision tree) algorithms used.

6 DISCUSSION OF RESULTS

These results were achieved using a set performance evaluation by class, notably; Decision Tree achieved a better performance in the results when compared to Naïve Bayes in term of accuracy with 97.60% in Decision Tree, 83.40% in Naïve Bayes. This was shown in table 3.0

7 CONCLUSION AND FUTURE WORK

Recent study shows that, researchers have proposed various techniques to detect ransomware. Basically, major works done on dynamic ransomware detection are based towards limited feature space. Therefore, there is a need to search for new features for ransomware detection. This work provides an overview of the ransomware phenomenon and its impact on businesses, institutions, and individuals. It also explored the behavior of ransomware in a digital attack. Then also summarized and discussed related works done on static, dynamic, and hybrid ransomware detection.

To develop ransomware detection and classification model, analysis reports of ransomware variants from the literature and industry were surveyed.The behaviors of ransomware that can be converted to a feature set were identified. These features were identified using machine learning techniques (Naïve Bayes and Decission tree).

In this research, a preprocessed dataset that comprises of ransomware and benign files were used to develop the RW detection at runtime scheme. Benign is good ware, and RW is a special type of malware that keeps the data encrypted until a ransom is paid to the attacker. In the experiment, the two algorithms: decision tree and Naïve Bayes were used to detect the RW and benign files. The decision tree algorithm, performed well in terms of accuracy, sensitivity, specificity, and f1-measure. The experimental results shows that the presented malware classification's testing and training accuracy is 97.60%.

For future work, in order to keep up with the huge production of ransomware, new detection tech-niques has to be introduced that can detect unseen ransomware. Further research on this topic could be done finding out how changing the ransomware samples to more specific ransomware families will affect the results. Also, finding out how already built classifiers will show difference in performance. However, using other algorithms like deep learning approaches and blockchain approaches to prevent modern and future ransomeware types is also proposed.

8 REFERENCES

[1] Businesses Paid $301M to Ransomware Hackers Last Year, New Datto Study Finds, datto, 2017. [Online]. Available: https://www.datto.com/news/datto-releases-global-state-of-the-channel-ransomware-report. [Accessed: 05-Jul-2019]

[2] Kharraz A., Robertson W. ,. Balzarotti D. Bilge L, and Kirda E, (2015) "Cutting the gordian knot: A look under the hood of ransomware attacks," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 9148, pp. 3–24,

[3] Richard Winton, (2019), "Hollywood hospital pays $17,000 in bitcoin to hackers; FBI investigating -Los Angeles Times," Los Angeles Times2016,[Online].Available:https://www.latimes.com/business/technology/la-me-ln-hollywood-hospital-bitcoin-20160217-story.html. [Accessed: 06-Jul-2019].

[4] UToday, (2016). "University of Calgary makes significant progress to address systems issues | UToday. University of Calgary,". [Online]. Available: https://www.ucalgary.ca/utoday/issue/2016-06-08/university-calgary-makes-significant-progress-address-systems-issues. *[Accessed: 05-Jul-2019]*

[5] Shinde R., Van Der Veeken P, Van Schooten S, and Van Den Berg J, (2018) "Ransomware: Studying transfer and mitigation," Int. Conf. Comput. Anal. Secur. Trends, CAST 2016, no. July 2018, pp. 90–95, 2018

[6] Krzysztof Cabaj, Marcin Gregorczyk, and Wojciech Mazurczyk (2018). Software-definednetworking-based crypto ransomware detection using *http traffic characteristics.Computers & Electrical Engineering, 66:353–368.*

[7] O'Brien, D. (2017). Internet Security Report: Ransomware 2017. *Symantec, 17-19*

[8] Nieuwenhuizen, D. (2017). A behavioral-based approach to ransomware detection.

[9] Scaife N., Carter H., Traynor P., and Butler K. R. B., (2016) "CryptoLock (and Drop It): Stopping Ransomware Attacks on User Data," Proc. - Int. Conf. Distrib. Comput. *Syst., vol. -Augus, pp. 303–312.*

[10] Lau, H., Coogan, P., & Savage, K. (2015). Evolution of Ransomware. Symantec, 5-8. Retrieved February 2, 2019, fromhttp://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/the-evolution-of-ransomware.pdf

[11] Bhardwaj A., Avasthi V., Sastry H., and Subrahmanyam G. V. B (2016). CryptoLock (and Drop It): Stopping Ransomware Attacks on User Data. IEEE 36th International Conference on Distributed Computing Systems (ICDCS). doi:10.1109/icdcs.2016.46

[12] Mattias, W., Frick, J., Sjostrom, A., & Jarpe, E. (2017). A Novel Method for Recovery from Crypto Ransomware Infections. In 2nd IEEE International Conference on Computer and Communications (pp. 1354–1358). IEEE

[13] Zahra, A. and Munam, A. S. (2017).IoT Based Ransomware Growth Rate Evaluation and

Detection Using Command and Control Blacklisting, Proceedings of the 23rd International Conference on Automation & Computing, University of Huddersfield, Huddersfield, UK, 7-8

[14] Cerber. (2017, March 29). CerberRing: An In-Depth Exposé on CerberRansomware-as-a-Service. Retrieved from https://blog.checkpoint.com/2016/08/16/cerberring/

[15] Liska, A., & Gallo, T. (2017). Ransomware: Defending against digital extortion.Sebastopol: OReilly Media

[16] Vadim Kotov, Mantej Singh Rajpal(2014). In-Depth Analysis of the Most Popular Malware Families, Bromium, UnderstandingCrypto-Ransomware Report.

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

[18] Continella A., *et al*., (2016) "ShieldFS," Proc. 32nd Annu. Conf. Comput. Secur. Appl. - ACSAC '16, pp. 336–347.

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

[20] Chen Z.-G., Kang H.-S., Yin, and Kim S.-R., (2017) "Automatic Ransomware Detection and Analysis Based on Dynamic API Calls Flow Graph," *pp. 196–201*.

[21] Lanzi A, D., Balzarotti, and C. Kruegel, (2016) "Access Miner-Using System Centric Models For Malware Protection."

[22] Poudyal S., Subedi K. P., and Dasgupta D.,(2019) "A Framework for Analyzing Ransomware using Machine Learning," Proc. 2018 IEEE Symp. Ser. Comput. Intell. SSCI 2018, pp. 1692–1699.

[23] Karimi A., and Moattar M. H., (2017) "Android ransomware detection using reduced opcode sequence and image similarity," 2017 7th Int. Conf. Comput. Knowl. Eng. ICCKE 2017, vol. 2017-Janua, no. Iccke, pp. 229–234.

[24] Zheng, C.; Dellarocca, N.; Andronio, N.; Zanero, S.; Maggi, F., (2016) Greateatlon: Fast, static detection of mobile ransomware. In Proceedings of the International Conference on Security and Privacy in Communication Systems, Guangzhou, China, 10–12 October 2016; pp. 617–636.

[25] Kotthoff L, Thornton C, Hoos HH, Hutter F, Leyton-Brown K. Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA. The Journal of Machine Learning Research. 2017 Jan 1;18(1):826-30.

[26] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An update; SIGKDD Explorations, Volume 11, Issue 1. [Available Online: http :// www.cs. waikato.ac.nz/ ml/weka/index.html]

[27] Paul Rubens (2017), "common types of ransomeware" Esecurity planets, www.esecurityplanet.com. . [Accessed: 20-Jul-2020]