Enhanced Ransomware Detection Techniques using Machine Learning Algorithms

Dr.G.Usha,Associate Professor, Dept. of Computing Technology,SRM IST, KTR

Dr.P.Madhavan,Associate Professor, Dept. of Computing Technology,SRM KTR

Meenalosini Vimal Cruz,Dep of Information Tehnology Allen E. Paulson Georgia Southern University,

Mr.Vinoth N A S,Assistnt Professor, Dept. of Computing Technology ,SRM KTR

Ms.Veena, Assistant Professor, Dept. of Computing Technology, SRM KTR ,

Mrs. Maria Nancy, Assistant Professor, Dept. of Computing Technology ,SRM KTR

2021 4th International Conference on Computing and Communications Technologies (ICCCT) | 978-1-6654-1447-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCCT53315.2021.9711906

***Abstract*— A challenge that governments, enterprises as well as individuals are constantly facing is the growing threat of ransomware attacks. Ransomware is a type of malware that encrypts the user's files and then demands a huge sum of money from the user. This increasing complexity calls for more advancement and innovative ideas in defensive strategies used to tackle the problems. In this paper, firstly we discuss the existing research in the field of ransomware detection techniques and their shortcomings. Secondly, a juxtaposed study on various machine learning algorithms to detect ransomware attacks is compared for ransomware dataset. Thirdly, various behavioral data such as API Calls, Target files, Registry Operations, Signature, Network Accesses are collected for each ransomware and benign sample and the results are compared for various attributes to understand the behavior of the attack. In order to understand the behavior of the attack various Machine Learning Algorithms like KNN, Naïve Bayes, Random Forest, Decision Trees are used for training and testing the dataset.. Further optimization was done using hyper parameters to control the learning process. Finally, we have used the model(s) Accuracy, F1 Score, Precision and Recall to compare the results observed and suggesting how the roadmap for how efficiently the attacks can be prevented in future.**

Keywords—Ransomware, Detection, KNN, Random Forest, Naive Bayes, Decision Tree, Machine Learning, Locky, WannaCry

I. INTRODUCTION

In today's world, computers are an essential part of our lives. For instance, you can write a mail in a word processor, make changes to it, print multiple copies, and send it to somebody via email halfway across the world in just a few clicks. All these tasks would have taken days to accomplish before, but now they are completed within moments. The fact of the matter is, whether it is a Small Business, or an Enterprise, they all rely on computer systems today. Pairing this with the rise in cloud computing, rather poor cloud security, and smart devices everywhere contributes to a multitude of threats.

Malware is a term used to define any piece of software built with the intention of harming the computer system, user or the data. These malwares were first introduced during the early 1970s to cause disruptions and as much damage as possible.Types of malwares include Viruses, Worms, Trojans and Spywares, furthermore in this research paper we will be focusing on detection of ransomwares.

Ransomware is a type of malign software, which when given access to a user's computer system, will encrypt the data essentially rendering the system useless until the ransom amount is paid usually in the form of bitcoins. These software's are able to set up their working environment in the background, also creating a backdoor to the system[1-5] in the process to remain updated throughout the duration of attack. After the encryption is performed, the user is prompted with a ransom note (usually desktop wallpaper is set to the ransom note) with the details to the transaction ID and an intimidating note asking for money. Ransomware's first cases date back to 2005 in Russia. Since then these types of malwares have spread across the world, advancing in method of implementation and targeting of their victims. Crypto locker first surfaced in September 2013 and was advanced enough to target all versions of Windows. Victims would accidentally open these files and send them via email impersonating UPS, DHS[6][7] etc. Once activated, the malware would get to work and prompt with a timer of 72 hours and a ransom. In this paper, we have analysed some of the most popular ransomware attacks such as WannaCry, Locky, CryptoShield, Crysis, Win32.Blocker, Unlock26. These ransomware samples have been used from the ISOT Ransomware Dataset and contain trace files of the ransomwares.

We have used multiple machine learning algorithms and compared their results. The paper is divided into subsections as follows: Section II discusses related work on ransomware related research. The proposed methodology is discussed in Section III, and the implementation is discussed in the section IV. Section V discusses the results observed. Finally, with Section VI the paper is concluded and future directions are discussed.

II. RELATED WORK

There has been a surge in the use of machine learning approaches to detect and prevent ransomware attacks.[2] A detailed study was made by Martina Jose Mary.M, Usharani.S, Thirugnanam.P in the paper called "Detection and Deterrence

of Ransomware using Machine Learning Techniques: A survey". In this survey they studied the detailed working of a ransomware attack and the different types of ransomware. They analyzed the features like CPU user usage, system usage, RAM usage, receive packet and byte, send packets, send bytes, receive packets, receive bytes, and net flows of the dataset. They used various Machine learning algorithms such as KNN, Naïve Bayes, Random Forest, SGD, SVM, Logistic Regression, and Bayesian Network to get desired output. They showed the comparison of all the accuracies and results as well. But every algorithm has its own advantages and disadvantages, so the best suited model which can be handled by the system is to be chosen. We improve on this study further by using different machine learning algorithms. We used the behavior of the ransomware trace files to train and test the model.

A paper written by Ban Mohammed Khammas titled[3] Ransomware Detection using Random Forest Technique" is a static method of detection. They used a random forest classifier to detect and found out that a high accuracy can be achieved by changing the seed value to 1 and tree numbers 100. They also used Frequent Pattern Mining technique to directly extract the features from raw byte. This showed an increase in efficiency. Next paper is "A Survey on Detection Techniques for Cryptographic Ransomware[4]" by Eduardo Berrueta, Daniel Morato, Eduardo Magaña, Mikel Izal. They made an in depth survey that concentrates on various detection models. In comparison to the previous studies, they offer a survey on different ransomware families and propose a few detection algorithms.

In the paper "BitcoinHeist: Topological Data Analysis for Ransomware Detection on the Bitcoin Blockchain[5]" by Cuneyt Gurcan, Yitao Li, Yulia R. Gel, Murat Kantarcioglu , they used advanced data analytics techniques to find ransomware related transactions and bitcoin addresses. But using bitcoin technology has its own disadvantages of varying bitcoin transaction addresses.

In the paper[6] "Towards Resilient Machine Learning for Ransomware Detection" by Li Chen, Chih-Yuan Yang, Anindya Paul, Ravi Sahita , they have compared the flexibility and stability of Machine Learning algorithms for security. A case study was done on evaluation of resilience using the generative adversarial network. They emphasized on the necessity of improving machine learning based approaches before final deployment. We gained a lot of insight on the various strategies and using ML techniques. It helped us understand how ML models can be made more resilient and robust.

Lastly, papers[7]-[10] helped us understand the current literature existing on the subject. Its survey consisted of research done in malware detection, prevention and recovery of the system after the attack. It helped us understand the existing work and further scope of research in this field from various papers in a summarized manner.

III. Proposed Ransomeware Attack Methodology

In this section, first the ransomware attack[11]-[15] methodologies are discussed. After analyzing various machine learning algorithms from[16]-[20] various machine learning algorithms are proposed to detect ransomware attacks.

Ransomware has 5 stages in which it is able to attack the user's system. These 5 Stages are:

1. Distribution: Malware is made and distributed through the internet.

2. Infection: Malware enters the system through the help of social engineering attack or other malwares such as trojans.

3. Staging: In this stage, ransomware application tries to distribute itself into multiple folders or tries to create a working environment in the system.

4. Scanning: In this stage, the ransomware scans for files that may be of importance to the user.

5. Encryption: The ransomware after scanning the files has what it requires and starts the file encryption process

6. Payday: A Ransom note appears on the users screen usually in the form of text or wallpaper.

From the above, it is observed that Stage 3 and 4 are essential stages for the successful execution of any ransomware. Hence, by taking countermeasures before or during these stages to prevent it from causing major damages to the user's system. Using an efficient Machine Learning model will help in analyzing and detecting the movements of the malware much earlier. In this research, we are focusing on the optimum Machine Learning model that could be used for this application. We are concentrating mainly on four algorithms.

KNN (K - Nearest Neighbor): It is one of the easiest algorithms used for classification and regression. It is a non-parametric method, which is also known as the lazy learning model with local approximation.

- It uses data points that are most similar as a base to classify data points. The working of KNN can be briefly explained as:
  - ✓ Select K as the number of neighbors
  - ✓ Calculation of Euclidean distance of K number of neighbors
  - ✓ Select nearest neighbors as the calculations
  - ✓ Among these, calculate the number of data points in each categories
- Assign the newer data to the categories

The Euclidian distance between two point A1 and B1 are calculated using the formula,

$$d = \sqrt{[(x_2 - x_1)^2 + (y_2 - y_1)^2]}$$

Where, A1 and B2 are data points and (X1, Y1) and (X2, Y2) are their coordinates respectively.

Decision Tree: This type of algorithm uses supervised machine learning in which the data is split based on a parameter. Decision Nodes and Leaves are the two entities of a tree. The

below example in Fig.1 explains the decision tree process for ransomware attack.

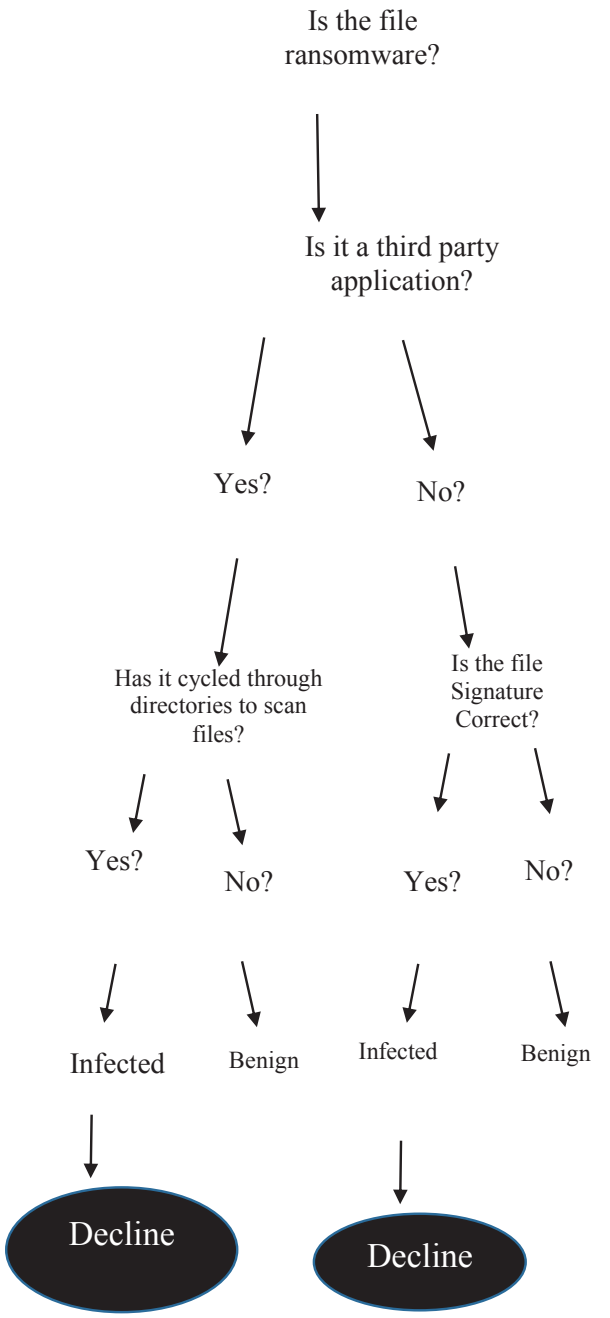

Fig.1. Ransomware Attack detection using Decision Trees

From the above example in Fig. 1 the following observations are made, "Is it a third party application?" and "Benign/Infected" are Decision nodes and leaves respectively. The working of Decision tree can be briefly explained as:

- ✓ Create a root node.
- ✓ Return 'Positive' value for leaf node, if all samples are positive.
- ✓ Return 'Negative' value for leaf node, if all samples are negative.
- ✓ Entropy for the Current State H(S) must be calculated
- ✓ Entropy must be calculated, for each attribute corresponding to its attribute 'x' which is represented as H(S, x).
- ✓ Attribute having the largest value of IG(S, x) must be selected.
- ✓ Attribute offering the greatest IG must be removed from the rest of the attributes.
- ✓ This must be repeated until we have no more attributes, or when the decision tree consists of only leaf nodes

Where, Information Gain (IG) is represented by IG(S,A), where in Set, change of entropy is denoted by S on a particular attribute A.

Random Forest: It is an easy-to-use algorithm that gives good results even without hyper-parameter tuning. It is optimal for both regression and classification tasks because of its flexible and diverse nature. It builds a "forest" that is a collection of decision trees trained using a method known as"bagging". It combines learning models that increases the results. The working of Random Forest can be briefly explained as:

- ✓ First random samples from the dataset are selected.
- ✓ A decision tree for every sample will be constructed.
- ✓ Prediction results from every decision tree are noted.
- ✓ Voting will be performed for every result that is predicted.
- ✓ Finally, we will select the most voted result as the final prediction result.

$$IG(S,A)=H(S)-H(S,A)$$

Alternatively,

$$IG(S,A)=H(S)-\Sigma P(x) * H(x)$$

mean and standard deviation of the points within each label.

- ✓ At every data point, the distance between z-score distance and each class-mean is calculated, i.e, distance from class mean divided by the standard

deviation.

- ✓ Prediction results from every decision tree are noted. Voting will be performed for every result that is predicted.
- ✓ Finally, we will select the most voted result as the final prediction result.

$$RFf_{ii} = \Sigma j \in all\ trees\ norm\ f_i\ ij/T$$

- ✓ RFfi sub(i)= Significance of feature 'i' by calculating from all the trees in the model.
- ✓ normfi sub(ij)= Significance of the normalized feature for 'i' in tree 'j'
- ✓ T = Number of total trees.

Gaussian Naïve Bayes: It is a group of supervised algorithms that use Bayes theorem for classification. It is an easy classification technique with high functionality. Gaussian Naïve Bayes uses Gaussian Normal Distribution. It is used for continuous data distribution. After calculating the probabilities for input values for each class using a frequency, we calculate the mean and standard deviation from the training data.

$$P(xi|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}}\ exp\left(\frac{(x_{i-}\ \mu_y)^2}{2\sigma_y^2}\right)$$

- ✓ We must assume the data given to us is defined by gaussian distribution having no independant dimensions nor co-variance.
- ✓ We can make the model fit by finding the mean and standard deviation of the points within each label.
- ✓ At every data point, the distance between z-score distance and each class-mean is calculated, i.e, distance from class mean divided by the standard deviation.

B. System Architecture

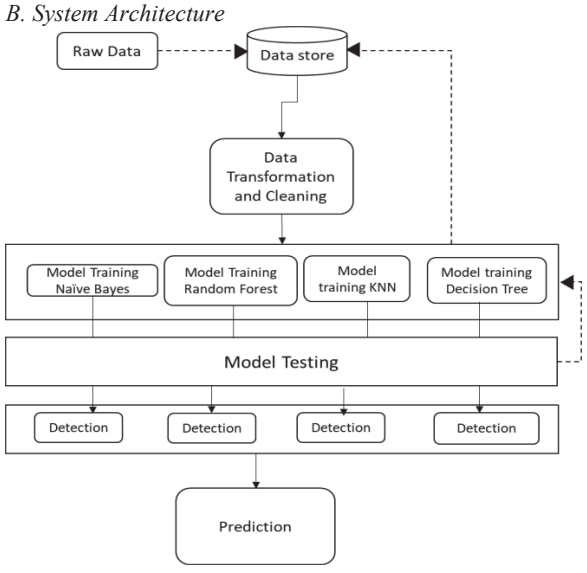

Fig. 2 - System Architecture

The above Fig. 2 explains the overall proposed system architecture for detecting ransomware attacks using various machine-learning algorithms.

- ✓ Raw Data: This consists of unorganized raw data. It contains trace files from existing ransomwares about their behavioural data from the ISOT Ransomware Detection Dataset. The file format is '.json' and must be converted to '.csv' format to make it compatible with the models
  - ✓ Data Store : This module converts the '.json' files to '.csv' files and stores them for further use.
  - ✓ Data Transformation and Cleaning: This module is used to clean the data by removing NaN values, converting the data using label encoding to convert Learning Models (KNN, Decision Tree, Naive Bayes, Random Forest) that will process the data fed in from the training dataset and get trained. These models are tuned accordingly using Hyperparameters for each model and optimizing it to give the best accuracy possible.
  - ✓ Validation : This module will use the trained model on the testing dataset, which will allow us to validate our results. Along with Model accuracy, it will also give us the precision, recall, f1 score and the confusion matrix for the respective models.

C. Dataset

We used the ISOT Ransomware Detection Dataset from the ISOT Research Lab, University of Victoria. It is a behaviour data of a collection of ransomware and benign samples. They were obtained from Virustotal1 under academic license along with several samples from anti-malware companies. The dataset consists of a total of 669 ransomware samples from ransomware families that are most widely used. The size of the entire dataset on disk is 428 GB. Apart from the ransomware samples, the dataset also includes 103 benign most used windows applications. Fig.3 explains the sample of top level structure nodes.

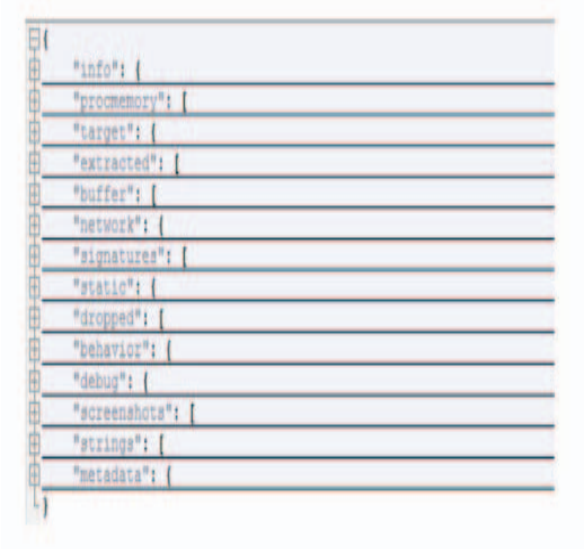

Fig. 3 - Sample Report Top Level Structure

A malware analysis sandbox called the cuckoo was installed in the host machine. It is a free open-source software tool that automates the analysis of malicious files. The dataset contains information about the analysis task, duration analysis, various memory regions, established network connections, processes created by sample, system API calls during the initial analysis, arguments, return values, strings extracted from the binary file of the analysed sample, different operations on file system and Windows registry. Next the evaluation methods are discussed for implementing the proposed model.

D. Evaluation

In order to implement the proposed scenario, ISOT Ransomware Detection Dataset from the ISOT Research Lab, University of Victoria are used. This dataset contains the behavior data of a collection of ransomware and benign samples. They were obtained from Virustotal under academic license along with several samples from anti-malware companies. The dataset consists of 669 ransomware samples from ransomware families that are most widely used. The size of the entire dataset on disk is 428 GB. Apart from the ransomware samples, the dataset also includes 103 benign most used windows applications.

The dataset contains information about the analysis task, duration analysis, various memory regions, established network connections, processes created by sample, system API calls during the initial analysis, arguments, return values, strings extracted from the binary file of the analyzed sample, different operations on file system and Windows registry.

✓ Classification Accuracy - It is the correct number of predictions divided by the number of total input samples. It is given as

$$Accuracy = \frac{number\ of\ correct\ predicitons}{total\ number\ of\ predictions\ made}$$

✓ Confusion Matrix- The output is given in the matrix form and it describes the models complete performance. Where,

TP: True Positive TN: True Negative

FP: False Positive FN: False Negative

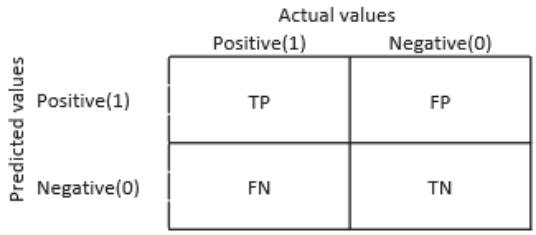

Fig. 4 - Confusion Matrix Representation

F1 score- Test's accuracy can be measured by it. It is also the Harmonic Mean between recall and precision. The score ranges from [0,1]. It tells how precise and robust your model is. Mathematically, it is given as -

$$F_1 = 2 * \frac{1}{(1|precision)+(1|recall)}$$

F1 Score finds the equilibrium between Recall and Precision.

IV. Results

The following results were observed:

- KNN achieved an accuracy of 94.7% but as it is a lazy learning algorithm its execution cost will tend to be greater on bigger datasets.

KNN
Accuracy = 0.9475524475524476
Confusion matrix =
[[77 0]
[15 194]]

	precision	recall	f1-score	support
0	0.84	1	0.91	77
1	1	0.93	0.96	209
accuracy			0.95	286
macro avg	0.92	0.96	0.94	286
weighted avg	0.96	0.95	0.95	286

Fig. 5 - KNN Results

- Decision Tree achieved an accuracy of 91.1%

Decision Tree
Accuracy = 0.9125874125874126
Confusion matrix =
[[52 25]
[0 209]]

	precision	recall	f1-score	support
0	1	0.68	0.81	77
1	0.89	1	0.94	209
accuracy			0.91	286
macro avg	0.95	0.84	0.87	286
weighted avg	0.92	0.91	0.91	286

Fig. 6 - Decision Tree Results

- Naive Bayes achieved an accuracy of 90.2%

Naïve Bayes
Accuracy = 0.9020979020979021
Confusion matrix =
[[76 1]
[27 102]]

	precision	recall	f1-score	support
0	0.74	0.99	0.84	77
1	0.99	0.87	0.93	209
accuracy			0.9	286
macro avg	0.87	0.93	0.89	286
weighted avg	0.93	0.9	0.91	286

Fig. 7 - Naive Bayes Results

Random Forest
Accuracy = 0.972027972027972
Confusion matrix =
[[74 3]
[5
204]]

	precision	recall	f1-score	support
0	0.94	0.96	0.95	77
1	0.99	0.98	0.98	209
accuracy			0.97	286
macro avg	0.96	0.97	0.96	286
weighted avg	0.97	0.97	0.97	286

Fig. 8 - Random Forest Results

Random forest achieved 96 percent accuracy.

V. Conclusion and Future Work

Over the years, researchers have helped develop various strategies to aid in detecting and preventing malware attacks. Signature based detection is one of the most popular strategies used, but is challenged by newer and advanced malwares being churned out by hackers. Hence, an efficient and accurate dynamic based machine learning approach is required.

As we can conclude, from the results shown in the previous section:

- Random Forest is able to perform with higher accuracy compared to the other three models (Naive Bayes, KNN, Decision Tree). This model is able to distinguish between benign and infected files efficiently using the traces provided during the training and validation process. Though, the Random Forest model has an expensive cost of estimation trees, which is slightly heavier on the system, but provides a more stronger and stable accuracy.
- Another good alternative to the random forest model is the KNN model, as it has also performed exceptionally well. Though, this KNN model is more suitable with smaller datasets due to high real time execution cost but allows for on the fly addition of data without affecting the accuracy of the system.

The future scope of this research is to implement the system in real time and test it thoroughly. This system can be further improved on, if paired with a signature based static method and will help detect malicious files earlier if code was not modified to bypass the signature based approach, but further research is required on that.

VI. Reference

[1]. Jethva, B., Traoré, I., Ghaleb, A., Ganame, K. and Ahmed, S., 2020. Multilayer ransomware detection using grouped registry key operations, file entropy and file signature monitoring. Journal of Computer Security, 28(3), pp.337-373.

[2]. Usharani, S. and Sandhya, S.G., 2020, July. Detection of ransomware in static analysis by using Gradient Tree Boosting Algorithm. In 2020 International Conference on System, Computation, Automation and Networking (ICSCAN) (pp. 1-5). IEEE.

[3]. Khammas, B.M., 2020. Ransomware Detection Using Random Forest Technique. ICT Express, 6(4), pp.325-331.

[4]. Berrueta, E., Morato, D., Magana, E. and Izal, M., 2019. A survey on detection techniques for cryptographic ransomware. IEEE Access, 7, pp.144925-144944.

[5]. Akcora, C.G., Li, Y., Gel, Y.R. and Kantarcioglu, M., 2019. BitcoinHeist: Topological data analysis for ransomware detection on the bitcoin blockchain. arXiv preprint arXiv:1906.07852.

[6]. Chen, L., Yang, C.Y., Paul, A. and Sahita, R., 2018. Towards resilient machine learning for ransomware detection. arXiv preprint arXiv:1812.09400.

[7]. Yang, C.Y. and Sahita, R., 2020. Towards a Resilient Machine Learning Classifier--a Case Study of Ransomware Detection. arXiv preprint arXiv:2003.06428.

[8]. Yang, C.Y. and Sahita, R., 2020. Towards a Resilient Machine Learning Classifier--a Case Study of Ransomware Detection. arXiv preprint arXiv:2003.06428.

[9]. Alsoghyer, S. and Almomani, I., 2019. Ransomware detection system for Android applications. Electronics, 8(8), p.868.

[10]. Kok, S.H., Abdullah, A., Jhanjhi, N.Z. and Supramaniam, M., 2019. Prevention of crypto-ransomware using a pre-encryption detection

algorithm. Computers, 8(4), p.79.
[11]. Agarwal, S., Tyagi, A. and Usha, G., 2020. A Deep Neural Network Strategy to Distinguish and Avoid Cyber-Attacks. In Artificial Intelligence and Evolutionary Computations in Engineering Systems (pp. 673-681). Springer, Singapore.
[12]. Butt, U.J., Abbod, M.F. and Kumar, A., 2020. Cyber threat ransomware and marketing to networked consumers. In Handbook of research on innovations in technology and marketing for the connected consumer (pp. 155-185). IGI Global.
[13]. Takeuchi, Y., Sakai, K. and Fukumoto, S., 2018, August. Detecting ransomware using support vector machines. In Proceedings of the 47th International Conference on Parallel Processing Companion (pp. 1-6).
[14]. Kolodenker, E., Koch, W., Stringhini, G. and Egele, M., 2017, April. Paybreak: Defense against cryptographic ransomware. In Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security (pp. 599-611).
[15]. Continella, A., Guagnelli, A., Zingaro, G., De Pasquale, G., Barenghi, A., Zanero, S. and Maggi, F., 2016, December. ShieldFS: a self-healing, ransomware-aware filesystem. In Proceedings of the 32nd Annual Conference on Computer Security Applications (pp. 336-347).
[16]. Ahmad, I., Basheri, M., Iqbal, M.J. and Rahim, A., 2018. Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. IEEE access, 6, pp.33789-33795.
[17]. Meidan, Y., Bohadana, M., Shabtai, A., Ochoa, M., Tippenhauer, N.O., Guarnizo, J.D. and Elovici, Y., 2017. Detection of unauthorized IoT devices using machine learning techniques. arXiv preprint arXiv:1709.04647.
[18]. Schultz, M.G., Eskin, E., Zadok, F. and Stolfo, S.J., 2000, May. Data mining methods for detection of new malicious executables. In Proceedings 2001 IEEE Symposium on Security and Privacy. S&P 2001 (pp. 38-49). IEEE.
[19]. D. Bhalla, Random Forest tutorial, 2014, Available:http://www.listendata.com/2014/11/random-forest-with-r.html.
[20]. Virus Total - Intelligence Search Engine, http://www.virustotal.com.