ISSN No:-2456-2165

Evaluation of Supervised Machine Learning Classifiers for Detecting Ransomware based on Naïve Bayes, SVM, KNN, C 4.5, and Random Forest Algorithms

Mohammed A. Saleh
Computer Department, College of Science and Arts in Ar Rass
Qassim University
Ar Rass, Kingdom of Saudi Arabia (KSA)

Abstract:- Nowadays, the wide spread of ransomware poses a destructive damage to the end users, which need to be tackled and treated properly to classify them and keep them away. Since the attributes and features of ransomware samples are extremely changeable, an automated analysis using machine learning algorithms is applied in order to handle the rapid changes of ransomware attributes features. In this paper, supervised machine learning classifiers (algorithms) such as Naïve Bayes, SVM, kNN, C 4.5, and Random Forest are evaluated for detecting ransomware. Several recent ransomware samples are collected, and their attributes and features are extracted and tabulated to construct training and testing datasets. Then, the datasets are evaluated and analyzed using Weka software for each classifier in three different modes, namely 10-fold cross-validation mode, 66.0% train split mode, and supplied test set mode. The best result for detecting ransomware is achieved by kNN classifier in 66.0% train split mode, which correctly classified 87.5% of instances, and therefore, the research suggests it for detecting ransomware.

Keywords:- *Malware Analysis; Supervised Machine Learning Algorithms; Ransomware Detection;*

I. INTRODUCTION

Ransomware is a destructive software and special kind of malware, which is utilized mostly for generating illegal money by extorting victims for paying a ransom in order to get and recover back their encrypted (encoded) files and documents that are encrypted by unauthorized parties like attackers [1][2]. According to [3], ransomware damage costs were forecasted to surpass 5 Billion U.S. Dollars in 2017. Cyber Risk Management Project (CyRiM) estimated a global ransomware attack could cause 200 Billion U.S. Dollars, with 89 Billion U.S. Dollars to American economic damages in 2019 [4]. Herjavec Group predicted cybercrime would cost 6 Trillion US Dollars by 2021 [5].

Due to the frequent changes of ransomware attributes and features [6], which come up with new ransomware variants [2][7][8], machine learning algorithms are preferred for evaluation and analysis for detecting ransomware in such cases [9], and therefore; this research uses them for the sake of ransomware detection. Machine learning (ML) is the science that gives systems the ability to automatically learn and continuously improve from experience without being explicitly programmed. Machine learning (ML) is effectively being used for data analysis [2][10][11].

Broadly, machine learning contains three categories, namely supervised machine learning, unsupervised machine learning, and reinforcement machine learning [12][13][14][15]. Supervised machine learning is used in cases when a certain instances of dataset, training set, is labeled, but it needs to predict for other instances, testing set [14][16][17]. Unsupervised learning is used where need to discover implicit relationships in a given unlabeled dataset [17][18]. Reinforcement machine learning is used the machine is trained to make specific decisions [14][19].

This research utilizes the supervised machine learning category in order to classify the dataset of the collected samples, and therefore to detect ransomware accordingly. The supervised machine learning is applied based on algorithms for classifying data set, thus they are called classifiers. The research uses a classifier term as a synonym for an algorithm. The algorithms, or classifiers, of the supervised machine learning are Naïve Bayes, SVM, kNN, C 4.5, and Random Forest.

The rest of this paper is organized as follows: sections 2 discusses the literature review of previous related research works. Section 3 identifies and describes the relevant definitions and theories. Section 4 introduces a simplified methodology for evaluating supervised machine learning classifiers in order to detect ransomware. Section 5 presents the results and discusses their analyses accordingly. Finally, section 6 presents the conclusion and future work.

II. LITERATURE REVIEW ON METHODS FOR DETECTING RANSOMWARE

Previous Related Research Works

A research is conducted by [31] for analysing WannaCry ransomware in order to discover and extract behavioural indicators and essential Indicators of Compromise (IOCs). In addition, the research utilized Yara tool to create customized patterns for the previous missions. However, the research focused only on the dynamic analysis of WannaCry ransomware, and therefore it is only valid for WannaCry ransomware, which fails to detect the other types of ransomware.

Another research is done by [32] for detecting ransomware through applying Long-Short Term Memory (LSTM) networks in order to the extract API calls from the log of modified sandbox environment. The gap of this research is that it analyses ransomware under victim's operating systems, which means inevitable harmful infection.

A further research [33] proposed a new method to detect and block ransomwares by analysing the file operation procedure of the operating system on user's device and applying an access control scheme to the file operation procedure. These white-listed records consist of authorized access controls of objects, which are file types, to its corresponding equivalent subjects, which are programs and processes. However, this countermeasure can by compromised and bypassed by zero-day vulnerability techniques, since it launches a dropper process from spawned authorizer program, which inevitably will have the authorized access controls.

An extra research [34] introduced a novel mothed for recovery from crypto ransomware infections by renaming the system tool that handles shadow copies, so that it is likely to recover from infections from the most common Ransomwares. In addition, it developed and packaged easy to use script for this mission. The idea of this research depends on Shadow Copy of Windows operating system to recover from ransomware attacks. Nonetheless, Shadow Copy need to be activated, since it is not activated by default, and it need to be authorized only to the system administrator, as well. Also, it suffers from precise synchronization that loses recent updates.

A researcher [35] investigated methods to implement a honeypot to detect ransomware activity. It selected two options: the File Screening service of the Microsoft File Server Resource Manager feature and EventSentry to manipulate the Windows Security logs. Additionally, it developed a staged response to attacks to the system along with thresholds when there were triggered. However, this research analyses logs of Honeypot to detect ransomware, since it cannot be applied at the production environment, and unavoidable harmful attacks will occur to victim's systems, thus it provides reactive defence mechanism.

A Research [36] proposed a model for ransomware detection and prevention by analysing nonstructured data stored in Ecuadorian control and regulatory institution (EcuCERT) logs in order to detect behaviours' patterns of the main vulnerabilities related to ransomware that are found in Microsoft Windows Systems, and to use machine learning techniques. It utilized methods for selecting important variables of the Logs to decide which features best act the data that makes up the threat. This research study analyses ransomware based on logs of various systems. Although this research claimed providing a proactive protection from ransomware, it fails to achieve that, since it depends on logs that could be manipulated by attackers, which means inevitable harmful infection.

Additional research [37] implemented data mining techniques in order to correlate multi-level code components, which are derived from reverse engineering process, for finding unique association rules to identify ransomware families. This research leveraged a hybrid approach, which consists of static and dynamic analysis, in order to tackle forensic analysis of ransomware. The approach applied a reverse engineering based on assembly instruction level, libraries used in PE file structure level, and function calls used in the libraries. It calculated a Cosine Similarity between benign software and ransomware in order to detect ransomware, since they use distinct assembly instructions. However, this merit can be bypassed, since attackers play with assembly codes. In addition, it uses association rule mining to predict ransomware through their common used DLL libraries and Function Calls. Nonetheless, predicting ransomware through DLL libraries and functions can be bypassed because attackers replace their calls with a raw assembly codes.

A different research [38] proposed a new framework named 2entFOX to detect high survivable ransomwares (HSR). First, it analysed Windows operating System ransomwares' behaviour to discover appropriate features with high accuracy of detection and low rate of false positive alerts. After that, it applied Bayesian belief network. However, all of these features can be forged and decoyed by attackers. Such as of these features: access to cryptographic libraries, specific directories access key words, targeted files search key words, and abnormal access to the paths and files.

A research [2] presented an approach based on Machine Learning, which utilizes integrated method that consists of a combination of static and dynamic analysis in order to detect ransomware.

Nonetheless, attackers can counterfeit all of these features. Such as of these features: Function Length Frequency (FLF), Printable String Information (PSI), API function calls, Registry Key Operations, Process Operations, and Network Operations.

III. SUPERVISED MACHINE LEARNING NAÏVE BAYES, SVM, KNN, C 4.5, AND RANDOM FOREST ALGORITHMS

A supervised machine learning is used in cases when a certain instances of dataset, called training set, is labeled, but it needs to predict for other instances of dataset, namely testing set [12][13][16][17][15]. The next subsections cover supervised machine learning algorithms (classifiers) in more details.

A. Naïve Bayes Machine Learning Algorithm (Classifier)

Define abbreviations and acronyms the first time they are Naive Bayes algorithm (classifier) is a probabilistic machine learning classification algorithm, which relies on Bayes Theorem to calculate the probabilities for every event based on the prior knowledge of any event that related to the former one, and then it picks out the outcome with highest probability [12][20]. The name Naive is used because the event is independent of each other, which means changing event value does not immediately influence or change the other event values. The Bayes Theorem formula is displayed in (1) [12][20-21]:

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)} \quad (1)$$

where:
P (A | B): The Probability of A when B is given.
P (B | A): The Probability of B when A is given.
P (A): Probability of A.
P (B): Probability of B.

B. A Support Vector Machine (SVM) Machine Learning Algorithm (Classifier)

Support Vector Machine (SVM) is a supervised machine learning classification and regression algorithm (classifier) [22]. SVM algorithm (classifier) plots data items in n-dimensional space, which consists of n features, with the value of a particular coordinate. Then, it performs a classification by calculating the hyperplane that distinguish the two classes properly according to the following SVM formula (2) [13][22-24]:

$$w \,.\, x - b = 0 \quad (2)$$

for hyperplane, w: wight, x: varabile vecor, b: bias

$$w \,.\, x - b \geq 0 \text{ , for one class}$$
$$w \,.\, x - b < 0 \text{ , for another class}$$

C. k-Nearest Neighbors Algorithm (kNN) Machine Learning Algorithm (Classifier)

k-Nearest Neighbors (kNN) algorithm (classifier) is a non-parametric and lazy supervised machine learning classification algorithm [13][25]. kNN algorithm (classifier) classifies new dataset items based on a similarity measure, which uses one of distance functions, namely Euclidean Distance (3), Manhattan Distance (4), Minkowski Distance (5), and Hamming Distance (6), as follow [13][25]:

$$\text{Euclidean Distance} = \sqrt{\sum_{i=1}^{k}(x_i - y_i)^2} \quad (3)$$

$$\text{Manhattan Distance} = \sum_{i=1}^{k}|x_i - y_i| \quad (4)$$

$$\text{Minkowski Distance} = \left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q} \quad (5)$$

$$\text{Hamming Distance } (D_H) = \sum_{i=1}^{k} |x_i - y_i| \text{ , where } D_H = 0 \text{ when } x = y,\ D_H = 1 \text{ when } x \neq y \quad (6)$$

D. A Decision Tree (C 4.5) Machine Learning Algorithm (Classifier)

Decision Trees (DTs) are a non-parametric supervised machine learning classification and regression algorithms. In a decision tree, a node in a tree represents an attribute or a feature, a branch represents a decision, and a leaf represents an outcome. C 4.5 algorithm (classifier) is the most famous Decision Tree (DT) algorithm, and it principally generates a decision tree where the classes are split on node based on the information gain. C 4.5 algorithm (classifier) utilizes the highest information gain of an attribute or a feature for the splitting criteria [26-27]. The information gain formula is presented in (7) as follows [27]:

$$IG(T, \alpha) = H(T) - H(T \mid \alpha) \quad (7)$$

where $H(T \mid \alpha)$: the conditional Entropy of T given the attribute α

E. Random Forest Machine Learning Algorithm (Classifier)

Random Forest is a supervised machine learning classification and regression algorithm (classifier). It simply an ensemble of decision trees, which constructs various decision trees and combines them together to come up with a more accurate prediction. Frequently, it utilizes bagging technique to train itself in order to overcome overfitting problems and to increase the overall results as well [26][28]. The bagging technique formula is displayed as follows [28-30]:

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b\,(\grave{x}) \quad (8)$$

IV. A SIMPLIFIED METHODOLOGY OF EVALUATING OF NAÏVE BAYES, SVM, KNN, C 4.5, AND RANDOM FOREST SUPERVISED MACHINE LEARNING CLASSIFIERS FOR DETECTING RANSOMWARE

First, the datasets that are used for evaluating Naïve Bayes, SVM, kNN, C 4.5, and Random Forest supervised machine learning classifiers (algorithms) for detecting ransomware are constructed from attributes and features of the benign and ransomware samples. This research collected 18 different ransomware from online databases of ransomware samples [39-40]. The research eliminated three ransomware samples, namely jigsaw, Rex, and Unnamed_0 because they are not standalone executable applications, which are not applicable to be analysed dynamically. In the

meanwhile, the research collected same number for 18 benign samples form fresh installation of Windows operating systems 32 and 64 bits. Fig. 1 depicts a simplified methodology of evaluating of Naïve Bayes, SVM, kNN, C 4.5, and Random Forest supervised machine learning classifiers for detecting Ransomware.

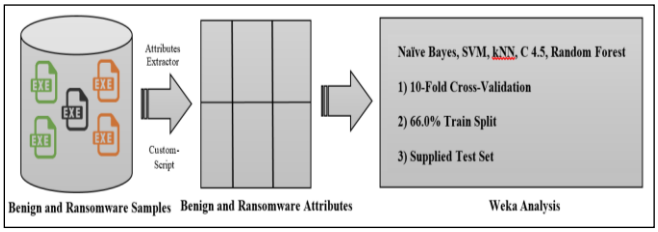

Fig 1:- A Simplified methodology of evaluating of Naïve Bayes, SVM, kNN, C 4.5, and Random Forest supervised machine learning classifiers for detecting ransomware

Second, a custom script, that is developed in this research as well, is used for extracting the attributes and features from the collected benign and ransomware samples. Listing 1 presents code of the custom script in Appendix 1. These attributes and features are tabulated in csv file, as presented in Table I, and then are passed to Weka software for the sake of analysis towards Naïve Bayes, SVM, kNN, C 4.5, and Random Forest supervised machine learning classifiers. The benign and ransomware samples are tabulated according to the five attributes, which involve Network Activity attribute, Code Execution attribute, Cryptography and Controls attribute, Device, Folder and File Functions attribute, and Process and Memory Functions attribute, as shown in Table I in Appendix 2.

Third, the dataset, which shown in Table I, is analyzed using Weka software [41] three times for each machine learning classifier (algorithm), and each time with a different mode. The analysis involves three different modes as follow: 10-fold cross-validation mode, 66.0% train split mode, and supplied test set mode. The whole results of analysis are presented and discussed in the next section.

V. RESULTS AND ANALYSIS

In this research, evaluating supervised machine learning classifiers such as Naïve Bayes, SVM, kNN, C 4.5, and Random Forest for detecting ransomware is conducted in three different modes, namely 10-fold cross-validation mode, 66.0% train split mode, and supplied test set mode, as follow:

A. Evaluation Of Supervised Machine Learning Naïve Bayes, SVM, KNN, C 4.5, AND Random Forest Classifiers for Detecting Ransomware In 10-Fold Cross-Validation Mode

First, in 10-fold cross-validation mode, Naïve Bayes classifier correctly classified 58.3333% of instances, while it incorrectly classified 41.6667% of instances. In addition, True Positive (TP) Rate, False Positive (FP) Rate, Precision, and Recall of Naïve Bayes classifier for detecting Ransomware are 0.333, 0.167, 0.667, and 0.333 respectably, and 0.833, 0.667, 0.556, and 0.833 respectably for detecting Benign. The average True Positive (Avg. TP) Rate equal 0.583, average False Positive (Avg. FP) Rate equal 0.417, average Precision equal 0.611, and average Recall equal 0.583. SVM classifier correctly classified 66.6667% of instances, whereas it incorrectly classified 33.3333% of instances. Besides, True Positive (TP) Rate, False Positive (FP) Rate, Precision, and Recall of SVM classifier for detecting Ransomware are 0.583, 0.250, 0.700, and 0.583 respectably, and 0.750, 0.417, 0.643, and 0.750 respectably for detecting Benign. The average True Positive (Avg. TP) Rate equal 0.667, average False Positive (Avg. FP) Rate equal 0.333, average Precision equal 0.671, and average Recall equal 0.667. kNN classifier correctly classified 70.8333% of instances, whereas it incorrectly classified 29.1667% of instances. Also, True Positive (TP) Rate, False Positive (FP) Rate, Precision, and Recall of kNN classifier for detecting Ransomware are 0.667, 0.250, 0.727, and 0.667 respectably, and 0.750, 0.333, 0.692, and 0.750 respectably for detecting Benign. The average True Positive (Avg. TP) Rate equal 0.708, average False Positive (Avg. FP) Rate equal 0.292, average Precision equal 0.710, and average Recall equal 0.708. C 4.5 classifier correctly classified 66.6667% of instances, whereas it incorrectly classified 33.3333% of instances. Further, True Positive (TP) Rate, False Positive (FP) Rate, Precision, and Recall of C 4.5 classifier for detecting Ransomware are 0.667, 0.333, 0.667, and 0.667 respectably, and 0.667, 0.333, 0.667, and 0.667 respectably for detecting Benign. The average True Positive (Avg. TP) Rate equal 0.667, average False Positive (Avg. FP) Rate equal 0.333, average Precision equal 0.667, and average Recall equal 0.667. Finally, Random Forest classifier correctly classified 66.6667% of instances, whereas it incorrectly classified 33.3333% of instances. Moreover, True Positive (TP) Rate, False Positive (FP) Rate, Precision, and Recall of Random Forest classifier for detecting Ransomware are 0.667, 0.333, 0.667, and 0.667 respectably, and 0.667, 0.333, 0.667, and 0.667 respectably for detecting Benign. The average True Positive (Avg. TP) Rate equal 0.667, average False Positive (Avg. FP) Rate equal 0.333, average Precision equal 0.667, and average Recall equal 0.667.

Fig. 2 presents evaluation of Naïve Bayes, SVM, kNN, C 4.5, and Random Forest classifiers for detecting ransomware in 10-fold cross-validation mode, while Fig. 3 shows the average True Positive (Avg. TP) Rate, the average False Positive (Avg. FP) Rate, the average Precision, and the average Recall of these classifiers. kNN classifier achieved the best result in the evaluation in 10-

ISSN No:-2456-2165

fold cross-validation mode, which correctly classified 70.8333% of instances.

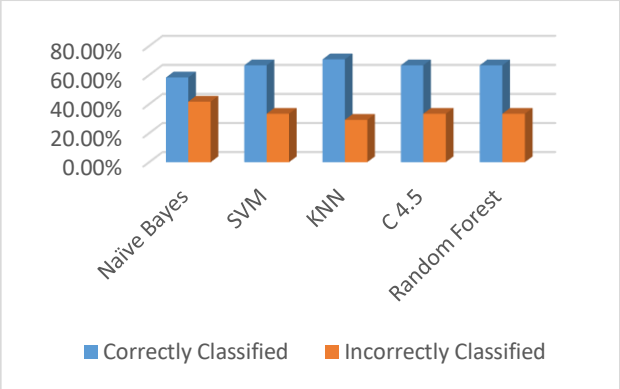

Fig 2:- Evaluation of Naïve Bayes, SVM, kNN, C 4.5, and Random Forest classifiers for detecting ransomware in 10-fold cross-validation mode

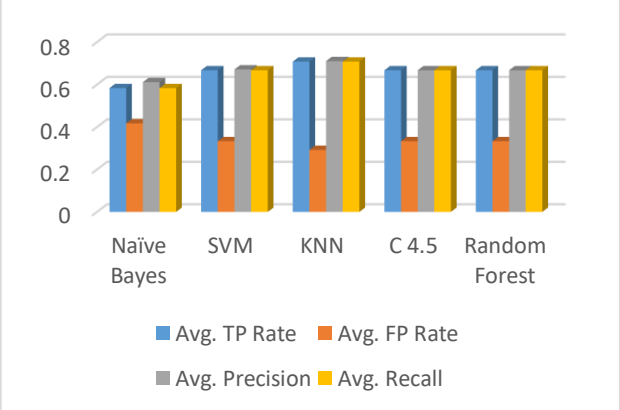

Fig 3:- The average true positive (Avg. TP) rate, average false positive (Avg. FP) rate, average precision, and average recall of Naïve Bayes, SVM, kNN, C 4.5, and Random Forest classifiers in 10-fold cross-validation mode

B. Evaluation Of Supervised Machine Learning Naïve Bayes, SVM, KNN, C 4.5, and Random Forest Classifiers for Detecting Ransomware In 66.0% Train Split Mode

Second, in 66.0% train split mode, Naïve Bayes classifier correctly classified 62.5% of instances, while it incorrectly classified 37.5% of instances. In addition, True Positive (TP) Rate, False Positive (FP) Rate, Precision, and Recall of Naïve Bayes classifier for detecting Ransomware are 0.800, 0.667, 0.667, and 0.800 respectably, and 0.333, 0.200, 0.500, and 0.333 respectably for detecting Benign. The average True Positive (Avg. TP) Rate equal 0.625, average False Positive (Avg. FP) Rate equal 0.492, average Precision equal 0.604, and average Recall equal 0.625. SVM classifier correctly classified 62.5% of instances, whereas it incorrectly classified 37.5% of instances. Besides, True Positive (TP) Rate, False Positive (FP) Rate, Precision, and Recall of SVM classifier for detecting Ransomware are 0.800, 0.667, 0.667, and 0.800 respectably, and 0.333, 0.200, 0.500, and 0.333 respectably for detecting Benign. The average True Positive (Avg. TP) Rate equal 0.625, average False Positive (Avg. FP) Rate equal 0.492, average Precision equal 0.604, and average Recall equal 0.625. kNN classifier correctly classified 87.5% of instances, whereas it incorrectly classified 12.5% of instances. Also, True Positive (TP) Rate, False Positive (FP) Rate, Precision, and Recall of kNN classifier for detecting Ransomware are 1.000, 0.333, 0.833, and 1.000 respectably, and 0.667, 0.000, 1.000, and 0.667 respectably for detecting Benign. The average True Positive (Avg. TP) Rate equal 0.875, average False Positive (Avg. FP) Rate equal 0.208, average Precision equal 0.896, and average Recall equal 0.875. C 4.5 classifier correctly classified 50% of instances, whereas it incorrectly classified 50% of instances. Further, True Positive (TP) Rate, False Positive (FP) Rate, Precision, and Recall of C 4.5 classifier for detecting Ransomware are 0.600, 0.667, 0.600, and 0.600 respectably, and 0.333, 0.400, 0.333, and 0.333 respectably for detecting Benign. The average True Positive (Avg. TP) Rate equal 0.500, average False Positive (Avg. FP) Rate equal 0.567, average Precision equal 0.500, and average Recall equal 0.500. Lastly, Random Forest classifier correctly classified 62.5% of instances, whereas it incorrectly classified 37.5% of instances. Moreover, True Positive (TP) Rate, False Positive (FP) Rate, Precision, and Recall of Random Forest classifier for detecting Ransomware are 0.800, 0.667, 0.667, and 0.800 respectably, and 0.333, 0.200, 0.500, and 0.333 respectably for detecting Benign. The average True Positive (Avg. TP) Rate equal 0.625, average False Positive (Avg. FP) Rate equal 0.492, average Precision equal 0.604, and average Recall equal 0.625.

Fig. 4 presents evaluation of Naïve Bayes, SVM, kNN, C 4.5, and Random Forest classifiers for detecting ransomware in 10-fold cross-validation mode, whereas Fig. 5 shows the average True Positive (Avg. TP) Rate, the average False Positive (Avg. FP) Rate, the average Precision, and the average Recall of these classifiers. kNN classifier achieved the best result in this mode, which correctly classified 87.5% of instances.

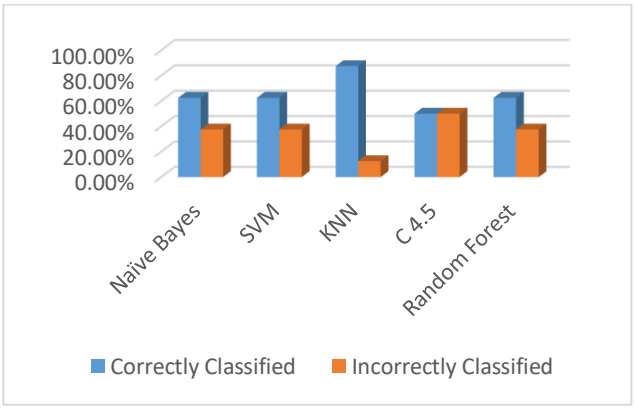

Fig 4:- Evaluation of Naïve Bayes, SVM, kNN, C 4.5, and Random Forest classifiers for detecting ransomware in 66.0% train split mode

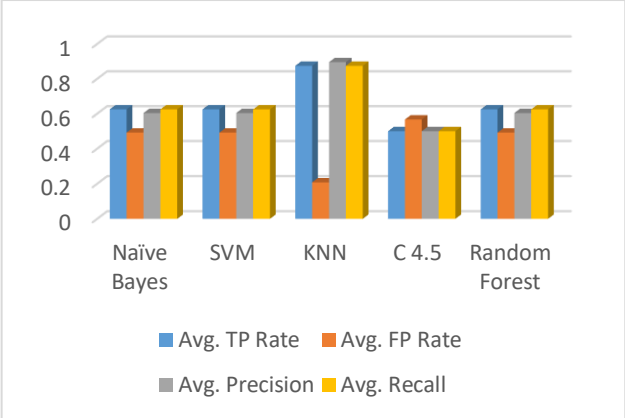

Fig 5:- The average true positive (Avg. TP) rate, average false positive (Avg. FP) rate, average precision, and average recall of Naïve Bayes, SVM, kNN, C 4.5, and Random Forest classifiers in 66.0% train split mode

C. Evaluation of Supervised Machine Learning Naïve Bayes, SVM, KNN, C 4.5, and Random Forest Classifiers for Detecting Ransomware in Supplied Test Set Mode

Third, in supplied test set mode, Naïve Bayes classifier correctly classified 66.6667% of instances, while it incorrectly classified 33.3333% of instances. In addition, True Positive (TP) Rate, False Positive (FP) Rate, Precision, and Recall of Naïve Bayes classifier for detecting Ransomware are 0.333, 0.000, 1.000, and 0.333 respectably, and 1.000, 0.667, 0.600, and 1.000 respectably for detecting Benign. The average True Positive (Avg. TP) Rate equal 0.667, average False Positive (Avg. FP) Rate equal 0.333, average Precision equal 0.800, and average Recall equal 0.667. SVM classifier correctly classified 66.6667% of instances, whereas it incorrectly classified 33.3333% of instances. Besides, True Positive (TP) Rate, False Positive (FP) Rate, Precision, and Recall of SVM classifier for detecting Ransomware are 0.333, 0.000, 1.000, and 0.333 respectably, and 1.000, 0.667, 0.600, and 1.000 respectably for detecting Benign. The average True Positive (Avg. TP) Rate equal 0.667, average False Positive (Avg. FP) Rate equal 0.333, average Precision equal 0.800, and average Recall equal 0.667. kNN classifier correctly classified 66.6667% of instances, whereas it incorrectly classified 33.3333% of instances. Also, True Positive (TP) Rate, False Positive (FP) Rate, Precision, and Recall of kNN classifier for detecting Ransomware are 0.333, 0.000, 1.000, and 0.333 respectably, and 1.000, 0.667, 0.600, and 1.000 respectably for detecting Benign. The average True Positive (Avg. TP) Rate equal 0.667, average False Positive (Avg. FP) Rate equal 0.333, average Precision equal 0.800, and average Recall equal 0.667. C 4.5 classifier correctly classified 75% of instances, whereas it incorrectly classified 25% of instances. Further, True Positive (TP) Rate, False Positive (FP) Rate, Precision, and Recall of C 4.5 classifier for detecting Ransomware are 0.500, 0.000, 1.000, and 0.500 respectably, and 1.000, 0.500, 0.667, and 1.000 respectably for detecting Benign. The average True Positive (Avg. TP) Rate equal 0.750, average False Positive (Avg. FP) Rate equal 0.250, average Precision equal 0.833, and average Recall equal 0.750. Finally, Random Forest classifier correctly classified 66.6667% of instances, whereas it incorrectly classified 33.3333% of instances. Moreover, True Positive (TP) Rate, False Positive (FP) Rate, Precision, and Recall of Random Forest classifier for detecting Ransomware are 0.333, 0.000, 1.000, and 0.333 respectably, and 1.000, 0.667, 0.600, and 1.000 respectably for detecting Benign. The average True Positive (Avg. TP) Rate equal 0.667, average False Positive (Avg. FP) Rate equal 0.333, average Precision equal 0.800, and average Recall equal 0.667.

Fig. 6 presents evaluation of Naïve Bayes, SVM, kNN, C 4.5, and Random Forest classifiers for detecting ransomware in 10-fold cross-validation mode, whilst Fig. 7 shows the average True Positive (Avg. TP) Rate, the average False Positive (Avg. FP) Rate, the average Precision, and the average Recall of these classifiers. C 4.5 classifier achieved the best result in the evaluation in supplied test set mode, which correctly classified 75% of instances.

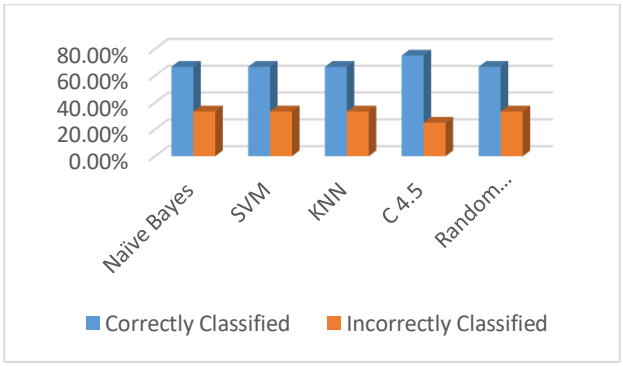

Fig 6:- Evaluation of Naïve Bayes, SVM, kNN, C 4.5, and Random Forest classifiers for detecting ransomware in supplied test set mode

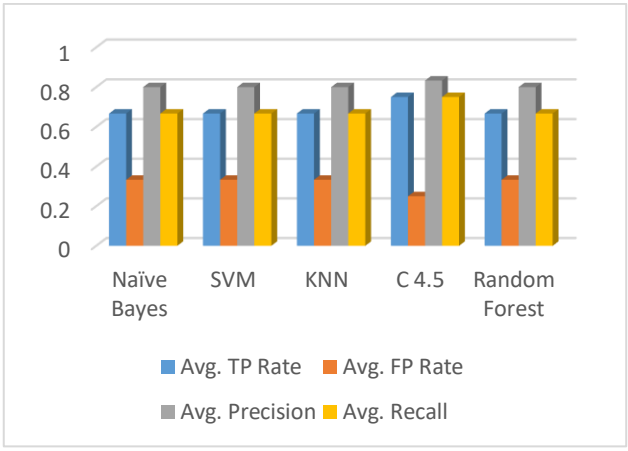

Fig 7:- The average true positive (Avg. TP) rate, average false positive (Avg. FP) rate, average precision, and average recall of Naïve Bayes, SVM, kNN, C 4.5, and Random Forest classifiers in supplied test set mode

Overall, kNN classifier in 66.0% train split mode achieved the best result in the evaluation, which correctly classified 87.5% of instances; therefore, the research suggests it for detecting ransomware. Fig. 8 presents the best results in the three modes.

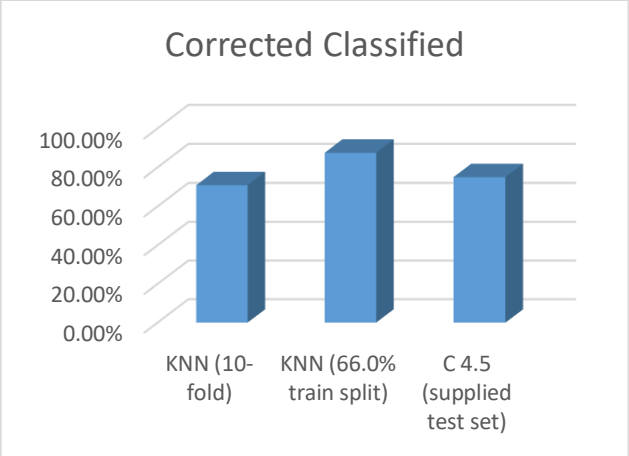

Fig 8:- The best achieved evaluation results in the three modes

VI. CONCLUSION

Due to the rapid changes of ransomware samples, an automated analysis and evaluation using machine learning algorithms is preferred to be applied in order to handle the such quick changes of ransomware attributes and features. This paper evaluated supervised machine learning algorithms (classifiers) like Naïve Bayes, SVM, kNN, C 4.5, and Random Forest in terms of ransomware detection. First, numerous recent benign and ransomware samples are collected from fresh installed Windows operating system and online databases [39][40], respectively. Then, the attributes and features of these collected samples are extracted and tabulated in order to construct training and testing datasets. After that, the datasets are analyzed and evaluated using Weka software for each classifier in three different modes, namely 10-fold cross-validation mode, 66.0% train split mode, and supplied test set mode. Based on the evaluation, the best result for detecting ransomware is achieved by kNN classifier in 66.0% train split mode, which correctly classified 87.5% of instances, and therefore; the research suggests using it for detecting ransomware.

REFERENCES

[1]. I. Kara and M. Aydos, "Static and Dynamic Analysis of Third Generation Cerber Ransomware," in 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT), 2018, pp. 12–17.
[2]. M. M. Hasan and M. M. Rahman, "RansHunt: A Support Vector Machines Based Ransomware Analysis Framework with Integrated Feature Set," 20th Int. Conf. Comput. Inf. Technol. (ICCIT), 22-24 December, 2017, pp. 22–24, 2017.
[3]. S. Morgan, "Global Ransomware Damage Costs Predicted To Exceed $5 Billion In 2017," May-2017.
[4]. T. C. R. M. (CyRiM) Project, "Bashe attack Global infection by contagious malware," 2019.
[5]. H. Group, "2019 Official Annual Cybercrime Report," 2019.
[6]. M. A. Saleh, "An Anatomy of Windows Executable File (EXE) and Linux Executable and Linkable Format File (ELF) Formats for Digital Forensic Analysis and Anti-Virus Design Purposes," Int. J. Adv. Res. Comput. Commun. Eng., vol. 7, no. 1, pp. 78–84, 2018.
[7]. K. P. Subedi, D. R. Budhathoki, and D. Dasgupta, "Forensic Analysis of Ransomware Families Using Static and Dynamic Analysis," in 2018 IEEE Security and Privacy Workshops (SPW), 2018, pp. 180–185.
[8]. D. Carlin, P. O'Kane, and S. Sezer, "Dynamic Opcode Analysis of Ransomware," in 2018 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), 2018, pp. 1–4.
[9]. S. Poudyal, K. P. Subedi, and D. Dasgupta, "A Framework for Analyzing Ransomware using Machine Learning," in 2018 IEEE Symposium Series on Computational Intelligence (SSCI), 2018, pp. 1692–1699.
[10]. E. Kirda, "UNVEIL: A large-scale, automated approach to detecting ransomware (keynote)," 2017 IEEE 24th Int. Conf. Softw. Anal. Evol. Reengineering, pp. 1–1, 2017.
[11]. T. K. Lengyel, S. Maresca, B. D. Payne, G. D. Webster, S. Vogl, and A. Kiayias, "Scalability, fidelity and stealth in the DRAKVUF dynamic malware analysis system," Proc. 30th Annu. Comput. Secur. Appl. Conf. - ACSAC '14, pp. 386–395, 2014.
[12]. Q. Liu, P. Li, W. Zhao, W. Cai, S. Yu, and V. C. M. Leung, "A survey on security threats and defensive techniques of machine learning: A data driven view," IEEE Access, vol. 6, pp. 12103–12117, 2018.
[13]. I. Firdausi, C. Lim, A. Erwin, and A. S. Nugroho, "Analysis of machine learning techniques used in behavior-based malware detection," Proc. - 2010 2nd Int. Conf. Adv. Comput. Control Telecommun. Technol. ACT 2010, pp. 201–203, 2010.
[14]. K. K. Mak, K. Lee, and C. Park, "Applications of machine learning in addiction studies: A systematic review," Psychiatry Res., vol. 275, pp. 53–60, 2019.
[15]. G. Shobha and S. Rangaswamy, "Chapter 8 - Machine Learning," in Computational Analysis and Understanding of Natural Languages: Principles, Methods and Applications, vol. 38, V. N. Gudivada and C. R. Rao, Eds. Elsevier, 2018, pp. 197–228.
[16]. S. Chadha and U. Kumar, "Ransomware: Let's fight back!," Proceeding - IEEE Int. Conf. Comput. Commun. Autom. ICCCA 2017, vol. 2017–Janua, pp. 925–930, 2017.
[17]. B. Wang, Y. Kong, Y. Zhang, D. Liu, and L. Ning, "Integration of Unsupervised and Supervised Machine Learning Algorithms for Credit Risk Assessment," Expert Syst. Appl., 2019.
[18]. C. Redondo-Cabrera and R. Lopez-Sastre, "Unsupervised learning from videos using temporal coherency deep networks," Comput. Vis. Image Underst., vol. 179, pp. 79–89, 2019.
[19]. C. You, J. Lu, D. Filev, and P. Tsiotras, "Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning," Rob. Auton. Syst., vol. 114, pp. 1–18, 2019.

[20]. H. Kim, J. Kim, J. Kim, and P. Lim, "Towards perfect text classification with Wikipedia-based semantic Naïve Bayes learning," Neurocomputing, vol. 315, pp. 128–134, 2018.
[21]. M. BAYGIN, "Classification of Text Documents based on Naive Bayes using N-Gram Features," in 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), 2018, pp. 1–5.
[22]. P. O'Kane, S. Sezer, K. McLaughlin, and E. G. Im, "SVM Training phase reduction using dataset feature filtering for malware detection," IEEE Trans. Inf. Forensics Secur., vol. 8, no. 3, pp. 500–509, 2013.
[23]. Y. An, S. Ding, S. Shi, and J. Li, "Discrete space reinforcement learning algorithm based on support vector machine classification," Pattern Recognit. Lett., vol. 111, pp. 30–35, 2018.
[24]. S. M. Kasongo and Y. Sun, "A Deep Learning Method with Filter Based Feature Engineering for Wireless Intrusion Detection system," IEEE Access, p. 1, 2019.
[25]. Z. Deng, X. Zhu, D. Cheng, M. Zong, and S. Zhang, "Efficient kNN classification algorithm for big data," Neurocomputing, vol. 195, pp. 143–148, 2016.
[26]. W. Wang, X. Wang, D. Feng, J. Liu, Z. Han, and X. Zhang, "Exploring permission-induced risk in android applications for malicious application detection," IEEE Trans. Inf. Forensics Secur., vol. 9, no. 11, pp. 1869–1882, 2014.
[27]. A. Zhandos and J. Guo, "An approach based on decision tree for analysis of behavior with combined cycle power plant," in 2017 International Conference on Progress in Informatics and Computing (PIC), 2017, pp. 415–419.
[28]. S. Fathi-Kazerooni, Y. Kaymak, and R. Rojas-Cessa, "Tracking User Application Activity by using Machine Learning Techniques on Network Traffic," in 2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), 2019, pp. 405–410.
[29]. R. M. Amir Latif, M. Umer, T. Tariq, M. Farhan, O. Rizwan, and G. Ali, "A Smart Methodology for Analyzing Secure E-Banking and E-Commerce Websites," in 2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST), 2019, pp. 589–596.
[30]. J. Schnebly and S. Sengupta, "Random Forest Twitter Bot Classifier," in 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), 2019, pp. 506–512.
[31]. D.-Y. Kao and S.-C. Hsiao, "The dynamic analysis of WannaCry ransomware," Int. Conf. Adv. Commun. Technolo gy(ICACT, pp. 159–166, 2018.
[32]. S. Maniath, A. Ashok, P. Poornachandran, and S. Jan, "Deep Learning LSTM based Ransomware Detection," Recent Dev. Control. Autom. Power Eng., vol. 3, 2017.
[33]. D. Y. Kim, G. Y. Choi, and J. H. Lee, "White list-based ransomware real-time detection and prevention for user device protection," 2018 IEEE Int. Conf. Consum. Electron. ICCE 2018, vol. 2018–Janua, pp. 1–5, 2018.
[34]. M. Wecksten, J. Frick, A. Sjostrom, and E. Jarpe, "A novel method for recovery from Crypto Ransomware infections," 2016 2nd IEEE Int. Conf. Comput. Commun. ICCC 2016 - Proc., pp. 1354–1358, 2016.
[35]. C. Moore, "Detecting ransomware with honeypot techniques," Proc. - 2016 Cybersecurity Cyberforensics Conf. CCC 2016, pp. 77–81, 2016.
[36]. J. A. H. Silva and M. Hernández-alvarez, "Large Scale Ransomware Detection by Cognitive Security," IEEE Second Ecuador Tech. Chapters Meet., pp. 1–4, 2017.
[37]. K. P. Subedi, D. R. Budhathoki, and D. Dasgupta, "Forensic Analysis of Ransomware Families Using Static and Dynamic Analysis," 2018 IEEE Secur. Priv. Work., pp. 180–185, 2018.
[38]. M. M. Ahmadian and H. R. Shahriari, "2entFOX: A framework for high survivable ransomwares detection," 13th Int. ISC Conf. Inf. Secur. Cryptology, Isc. 2016, pp. 79–84, 2016.
[39]. Y. Nativ, "theZoo project." [Online]. Available: https://github.com/ytisf/theZoo. [Accessed: 23-Mar-2019].
[40]. CrowdStrike, "reverse." [Online]. Available: https://www.reverse.it/. [Accessed: 02-Sep-2018].
[41]. N. Z. University of Waikato, "Weka 3." University of Waikato, New Zealand, 2019.

APPENDICS

A. Appendix 1

Listing 1 Benign and ransomware attributes and features extractor custom script code.

```
#!/bin/bash
INPUTEXEFILES=*.exe
INPUTFILES=*.txt
WORDS=(Internet connection network Http socket NetServer recv send Shell CommandLine cmd Resource icacls Crypt Cred Cert SecurityDescriptor EncodePointer User permission Privilege password Token Device Driver volume partition PhysicalDrive mbr boot Directory Folder File Temp FileAttributes Library Search Flush delete Reg Str print cpy cat MessageBox Clipboard Process Thread Event Mutex Servic task Timer ProcAddress Sleep exit mscoree.dll Heap VirtualAlloc FreeMemory memcpy malloc CriticalSection memset)
OUTPUTFILE=Counts
# First Part:
```

```
echo -e "First Part:"
for s in $INPUTEXEFILES
do
`strings -n 1 "$s" > "$s".txt`
echo "$s: convertion to strings ... done"
done
echo -e "\n"
# Second Part;
echo "Second Part:"
echo "Keywords Order: Internet connection network Http socket NetServer recv send Shell CommandLine cmd Resource icacls Crypt Cred Cert SecurityDescriptor EncodePointer User permission Privilege password Token Device Driver volume partition PhysicalDrive mbr boot Directory Folder File Temp FileAttributes Library Search Flush delete Reg Str print cpy cat MessageBox Clipboard Process Thread Event Mutex Servic task Timer ProcAddress Sleep exit mscoree.dll Heap VirtualAlloc FreeMemory memcpy malloc CriticalSection memset" > $OUTPUTFILE
echo "-------------------------------------" >> $OUTPUTFILE
for i in $INPUTFILES
do
echo "FN: $i" >> $OUTPUTFILE
echo "Counts: " >> $OUTPUTFILE
for j in ${!WORDS[*]}
do
`grep -i "${WORDS[j]}" "$i" | wc -l >> "$OUTPUTFILE"`
echo "$i$j ... Done"
done
done
```

B. Appendix 2

Name	Network Activity	Code Execution	Cryptography and Controls	Device, Folder and File Functions	Process and Memory Functions	Class
cerber.exe	2	7	15	106	40	Malware
cryptowall.exe	2	3	12	62	43	Malware
regedit.exe	3	14	21	139	26	Benign
winhlp32.exe	0	0	3	1	11	Benign
write.exe	1	6	4	3	12	Benign
locky.exe	3	0	10	54	14	Malware
mamba.exe	197	85	165	935	473	Malware
matsnu.exe	1	9	4	46	27	Malware
comp.exe	0	1	3	46	13	Benign
radaman.exe	13	8	30	118	42	Malware
satana.exe	1	0	3	10	2	Malware
DpiScaling.exe	1	3	6	4	11	Benign
chgport.exe	0	1	5	27	16	Benign
teslacrypt 1.exe	0	8	2	20	19	Malware
choice.exe	0	0	6	21	23	Benign
cliconfg.exe	1	1	5	8	11	Benign
teslacrypt 2.exe	16	7	17	117	52	Malware
vipasana 1.exe	10	14	9	127	44	Malware
cscript.exe	1	5	7	148	34	Benign
vipasana 2.exe	10	14	9	127	44	Malware
vipasana 3.exe	8	13	9	127	43	Malware
cmdl32.exe	54	15	8	216	48	Benign

wannacry.exe	0	11	13	62	34	Malware
wannacryPlus.exe	17	29	43	118	82	Malware
ComputerDefaults.exe	2	1	7	7	14	Benign
Defrag.exe	0	1	4	65	35	Benign
DisplaySwitch.exe	3	7	7	37	22	Benign
teslacrypt 3.exe	0	5	3	22	19	Malware
cmd.exe	4	7	12	80	49	Benign
control.exe	1	7	6	12	17	Benign
petya 2.exe	18	15	28	158	84	Malware
klist.exe	0	0	14	26	26	Benign
notepad.exe	3	8	7	71	30	Benign
petrwrap.exe	11	16	36	72	51	Malware
label.exe	0	0	3	38	12	Benign
petya 1.exe	80	21	59	426	119	Malware

Table 1:- The Extracted Attributes and Features of Ransomware and Benign Samples