Journal of Information Security and Applications 55 (2020) 102646

Contents lists available at ScienceDirect

Journal of Information Security and Applications

journal homepage: www.elsevier.com/locate/jisa

Evaluation metric for crypto-ransomware detection using machine learning

S.H. Kok *, A. Azween, NZ Jhanjhi *

Taylor's University, 47500 Subang Jaya, Selangor, Malaysia

ARTICLE INFO

Keywords:
Crypto
Encryption
Evaluation
Machine learning
Metric
Ransomware
Detection

ABSTRACT

Ransomware is a type of malware that blocks access to its victim's resources until a ransom is paid. Crypto-ransomware is a type of ransomware that blocks access to its victim's files by the use of an encryption algorithm. This encrypted file remains permanently blocked, even if the victim is able to remove the ransomware from the infected file. This has forced victims to pay the ransom demanded in exchange for a decryption key, although the decryption key provided is not guaranteed to work. To address this situation, we propose a pre-encryption detection algorithm (PEDA) for detecting crypto-ransomware prior to the occurrence of any encryption. The PEDA has two levels of detection. The first is a signature repository (SR) that identifies any matches of the signature with that of known ransomware. The second detection level uses a learning algorithm (LA) that can detect both known and unknown crypto-ransomware. LA uses a machine learning approach to train the predictive model using data from the application program interface (API). In order to understand PEDA functionality, LA is being evaluated using conventional metrics and unconventional metrics. Conventional metrics such as the true positive rate, accuracy, and precision can provide important performance indicator, but not comprehensive enough to assess the LA capability. Six new metrics had been proposed to provide greater insight. Based on the results, it can be concluded that LA had achieved its objective of detecting crypto-ransomware before the encryption is viable and that its performance is robust with a high net benefit.

1. Introduction

Ransomware, as the name implies, is malware that demands the payment of ransom from its victim. The first of three types of ransomware is called scareware, which tries to deceive its victim with a false threat. The other two types block access by the victim to their resources until a ransom is paid. This type of ransomware achieves its goal using one of two approaches, i.e., either by blocking access to the victim's system or encrypting the victim's files and data, as illustrated in Fig. 1. Ransomware that uses the first method is called locky-ransomware, and the latter is called crypto-ransomware. Crypto-ransomware is considered to be more destructive because the encrypted file remains inaccessible even after complete removal of the ransomware [1]. The encrypted file can only be restored to its normal state by the use of a specific decryption key. Depending on the encryption algorithm used, the use of any brute force method could take many years to recover the decryption key. As such, many corporations have been forced to pay the ransom in exchange for a decryption key, which may itself not always work [2].

Evaluation metrics are important tools used as performance indicators for a predictive model based on the machine learning approach [4]. However, conventional metrics mainly focus on providing benchmarks for predictive models but lack the capability to assess their performance regarding the likelihood of correct and wrong predictions, their optimum performance ranges, and the benefits of using a given predictive model.

Therefore the contribution of this paper is three folds, first is to use application program interface (API) before encryption happens as the data for analysis. The second contribution is the development of pre-encryption detection algorithm (PEDA) that has two levels of detection to improve the overall detection performance and accuracy. The first level of detection is called Signature Repository (SR) that uses signature matching for detection. The second level is called Learning Algorithm (LA) that uses a predictive model for detection. The third contribution is to propose six new metrics that can provide greater insight regarding the capability of the LA.

This paper is further organized such as the second section is the critical analysis of past literature and the research gap found. The third

This work is supported by Taylor's University through its TAYLOR'S PhD SCHOLARSHIP Programme.
* Corresponding authors.
E-mail addresses: koksimhoong@sd.taylors.edu.my (S.H. Kok), noorzaman.jhanjhi@taylors.edu.my (N. Jhanjhi).

https://doi.org/10.1016/j.jisa.2020.102646

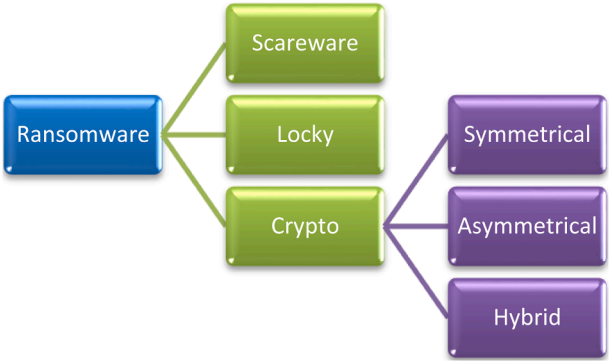

Fig. 1. Types of ransomware (adapted from [3]).

section provides the experimental setup for the implementation and performance evaluation of the PEDA. The fourth section discusses the results obtained from the experiment and its implication. The fifth section provides the conclusion by realigning the result with the proposed contribution set forth in this paper.

2. Related work

Machine learning is now being used to detect crypto-ransomware before the encryption process starts. The common practice when using a supervised machine learning method is to segregate the dataset in an 80:20 ratio, where 80% of the dataset is used to train the machine learning algorithm to produce a predictive model. This predictive model is then used to provide prediction results based on 20% of the dataset. The data segregation must be random but results in a similar ratio of ransomware and goodware.

The accuracy metric indicates the ratio of correct pre Based on the prediction results, the performance of the predictive model can be evaluated using a confusion matrix, as shown in Fig. 2. In this matrix, a true positive result indicates the number of predictions correctly predicted to be positive. A false positive indicates the number of predictions incorrectly predicted to be positive. A true negative indicates the number of predictions correctly predicted to be negative. A false negative indicates the number of predictions incorrectly predicted to be negative.

Based on the confusion matrix result, evaluation metrics can be derived to provide insight regarding the performance of the predictive model [5]. Some typical and popular conventional metrics used include accuracy, true positive rate (TPR), false positive rate (FPR), true negative rate (TNR), false negative rate (FNR), precision, and F-measure. dictions over the total number of predictions, as shown in Eq. (1). This metric determines how well the predictive model is able to make correct predictions.

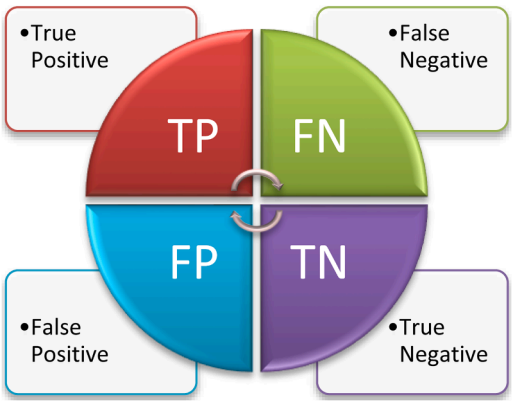

Fig. 2. Confusion matrix.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

The TPR indicates the ratio of predictions correctly predicted to be positive over the total number of actual positive conditions, as shown in Eq. (2). This metric determines how well the predictive model can correctly predict positive values. Other names for this metric include recall, sensitivity, and detection rate.

$$TPR = \frac{TP}{TP + FN} \tag{2}$$

The FPR indicates the ratio of predictions incorrectly predicted to be positive over the total number of actual negative conditions, as shown in Eq. (3). This metric determines the extent to which the predictive model incorrectly predicts positive values.

$$FPR = \frac{FP}{FP + TN} \tag{3}$$

The TNR, which is also called specificity, indicates the ratio of predictions correctly predicted to be negative over the total number of actual negative conditions, as shown in Eq. (4). This metric determines how well the predictive model can correctly predict negative values [6].

$$TNR = \frac{TN}{TN + FP} \tag{4}$$

The FNR indicates the ratio of predictions incorrectly predicted to be negative over the total number of actual positive conditions, as shown in Eq. (5). This metric determines the extent to which the predictive model incorrectly predicts negative values.

$$FNR = \frac{FN}{FN + TP} \tag{5}$$

The precision metric indicates the ratio of predictions correctly predicted to be positive over the total number of positive predictions, as shown in Eq. (6). This metric determines how much the predictive model can be trusted when the prediction is positive.

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

The F-measure, which is also called the F-score or the F1-score, is actually the mean of the TPR and precision, as shown in Eq. (7). This metric determines how well the predictive model can correctly predict positive values while taking into consideration both FN and FP.

$$F = \frac{2TP}{2TP + FP + FN} \tag{7}$$

The above metrics are commonly used to evaluate the performance of a predictive model produced by the machine learning method, but are these can be inadequate in certain cases. To be effective, evaluation metrics must provide insight regarding the strengths, weaknesses, and areas for improvement of a predictive model. Our objective in this paper is to propose new metrics for the evaluation of a predictive model used in ransomware detection.

Table 1 shows a summary of the evaluation metrics currently used in malware detection using the machine learning technique. As shown in the table, the most commonly used metrics in malware detection are TPR, followed by accuracy and precision. These metrics shows provide detection performance but does not show the capability of the predictive model. This is our first research gap for this paper.

Table 2 shows a summary of the achievement and limitation of the literature. Based on this, we can stress that none of the above provides early detection of crypto-ransomware using data prior to the encryption process. This stage is actually very crucial to avoid crypto-ransomware from encrypting files and hold it as a hostage to demand a ransom. This is the second research gap for this paper. The third research gap is obviously to develop the detection algorithm that can effectively detect

Table 1
Evaluation metric.

No.	Ref.	Metric						
		Acc	TPR	FPR	TNR	FNR	Prec	F-m
1	[7]							
2	[8]	√	√				√	
3	[9]							
4	[10]	√	√				√	√
5	[11]							
6	[12]		√	√				√
7	[13]		√	√				
8	[14]							
9	[15]							
10	[16]							
11	[17]							
12	[18]	√		√		√		
13	[19]							
14	[20]		√				√	
15	[21]	√			√		√	
16	[22]	√						
17	[23]	√						√
18	[24]		√				√	√
19	[25]	√	√	√				
20	[26]	√	√				√	√
21	[27]							
22	[28]		√				√	
23	[29]		√	√			√	√
24	[30]	√						
25	[31]		√				√	√
26	[32]		√	√				
Total		9	12	6	1	1	9	7

Acc – Accuracy.
Prec – Precision.
F-m – F – measure.

crypto-ransomware based on data from the second research gap.

2.1. Strengths

There are several important advantages in using the above metrics, the main one being ease of comparison of the performance results with those obtained in other research works. Using the same metric ensures the use of the same calculation formula, which enables a simple and direct comparison of the calculated values. The second advantage of using these metrics is their simple and easy formulas that enable fast and easy calculation. The third advantage is their good reflection of the degree of correctness and incorrectness of the predictions for both positive and negative results. This is important, especially in the evaluation of a supervised machine learning algorithm, which consists of pre-labelled positive and negative data. The fourth advantage of these metrics is that their values range between zero and one, which facilitates easy comparison and interpretation of the metrics.

2.2. Weaknesses

These metrics also have some weaknesses, the first being that none, except for accuracy, use all values in the confusion matrix. This means that they may not fully represent the results in the confusion matrix on which their formulas are based. The second weakness is the inability of the individual metrics to provide an indication of how well the predictive model can differentiate between positive and negative results. This is important, especially for malware detection, which requires that a prediction distinguishes between goodware and malware. The third weakness is that these metrics provide no indication regarding points of success and failure, i.e., a range at which the predictive model will perform at its optimal level. The fourth weakness is the lack of indication of the benefits of using a given predictive model.

Table 2
Critical analysis.

No	Ref	Achieved	Limitation	Research findings
1	[7]	Immediately blocked and notify for its removal	Future to test on other platforms such as Windows and Android	No guarantee ransomware will attack honey files
2	[8]	Software-Defined Networking (SDN) improves network protection with simple rules	Future to test on healthcare implant and other internet-connected gadgets	Did not try on goodware
3	[9]	8 API exists only in ransomware 4 API ransomware statistically significant 6 API frequency > 3 std dev	Nil	API differentiation, no actual detection mechanism
4	[10]	Best recall at 99.8% using Decision Tree 3-gram and 4-gram, K Nearest Neighbor 2-gram.	Cannot distinguish well crypto wall, locky and prevention according to accuracy for binary classification	10 minutes, API call comparison
5	[11]	Windows platform, detection by monitoring abnormal filesystem and registry activities. Android platform by controlling permissions.	Future to test on Linux and Mac platform	Suggestion, but no actual detection mechanism
6	[12]	Binary; F-measure, TPR, FPR, MCC Long Short Term-Memory (0.996, 0.992, 0, 0.986) Multi-class; TPR, FPR (0.972, 0.027)	Future to use other deep learning algorithms such as sequential discriminative training of the deep neural network, and ensemble deep neural network DNN, CNN, RNN	Monitor ransomware activities
7	[13]	AUC, Test Error, FPR, Detection Rate ElderRan (0.9949, 0.0238, 0.0161, 0.9634) VirusTotal (0.9993, 0.0561, 0.0000, 0.8530)	Cannot detect ransomware that waits for user action	10 seconds runtime limit
8	[19]	Malware only; homogeneity (0.767), completeness (0.609), v-measure (0.679), Mixture malware and legitimate operation; homogeneity (0.761), completeness (0.523), v-measure (0.620),	Limitation, fails on samples that do not interact with resources monitored by the sandbox	20 seconds runtime limit
9	[20]	VTCSandbox@8x runs 102s, Precision (98.6)	Nil	Fast detection of time-based malware
10	[21]	Most accurate is CW with C = 4.0 and n = 6, train (0.940), test (0.918)	Nil	Virtual clock, did not mention detection technique
11	[22]	Sophisticated Attacker KuafuDet (96.20)	Future work to use reinforcement technology to prevent APK from reverse-engineering	3min, add weight vector, balanced accuracy metric
12	[23]	500 data Accuracy w/o bigram (82%), Accuracy w/ RFA (92.9%), Combined (87.8%)	Future to use ensemble classifier and trigrams technique	Adversary detection, camouflage detector, FN metric
13	[24]			

(continued on next page)

Table 2 (*continued*)

No	Ref	Achieved	Limitation	Research findings
		Malware Operational Plot Review (MOPR) % Correct (93.76),	Future to use model phases of system behavior to detect the attack at an early stage.	Network packet analysis
14	[25]	Accuracy support vector machine (0.944)	Nil	5min runtime limit
15	[26]	Random Forest, accuracy Raw (97.43), IntF (98.78)	Future to test on application files such as multimedia, document processing, device drivers, etc.	Portable executable header
16	[27]	Able to detect author created Ransomware that can bypass two antiviruses	Nil	Portable executable
17	[28]	Able to detect the known and unknown type of Ransomware	An attacker may be able to detect the artificial environment or run at kernel level that can thwart monitoring of UNVEIL	Use author created Ransomware
18	[29]	J48 Decision Tree produces the best True Positive Rate of 97.1%	Future to develop real-time Ransomware detection using cloud-based ML classifiers	20min runtime limit
19	[30]	91% accuracy	Future to use Natural Language Processing (NLP) and spelling auto-correction	Depends on availability of ransom note
20	[31]	99% accuracy	Future to detect other malware such as botnet and rootkit	Structural Similarity Metric
21	[32]	The best result from Gradient Tree Boosting (GTB) with 98.25% TPR and 0.56% FPR	Evaluate in large scale real setup	Combine detection and preservation

3. Methodology

In this study, we collected crypto-ransomware samples from three sources, namely Old, VirusTotal, and theZoo. The Old source comprises samples used by the authors in [13], and VirusTotal and theZoo are two online-based repositories of malware samples available to the public. The obtained samples were analysed dynamically using the Cuckoo Sandbox analysis system to capture all the API requests in each sample. This information was extracted and converted into dataset format for machine learning training and testing of the LA from the PEDA. The test results were then evaluated using the proposed metrics.

3.1. Data

A total of 904 ransomware samples were ultimately collected from the three sources, as shown in Table 3 below. We used 491 ransomware samples (Old) and 942 goodware samples provided in [13]. Additional new ransomware samples were collected from online repositories available to the public, i.e., VirusTotal and theZoo, for a total of 357 ransomware samples from VirusTotal and 56 ransomware samples from theZoo.

In the data extraction phase, we generated three datasets, as listed in Table 4. Except for 91 ransomware samples, those from the Old dataset, used by the authors in [13], could not be processed by our Cuckoo Sandbox. This may have been due to our use of a different version of the program. The Full dataset contained all collected ransomware, both old and new. The third dataset, namely the pre-encryption (PE) dataset, consisted of ransomware samples that were characterized by API encryption from Windows. When a sample was identified as exhibiting encryption behaviour, the API data extraction process was stopped, as we only wanted data at the pre-encryption stage.

3.2. Pre-encryption detection algorithm (PEDA)

The PEDA for detecting crypto-ransomware uses the API at the pre-encryption stage to impede the encryption function of the ransomware. Successful detection of crypto-ransomware at the pre-encryption stage is important to prevent files from being rendered irrecoverable. Fig. 3 below shows the process flow of PEDA, which involves two levels of crypto-ransomware detection.

At the first level, signature matching is performed by comparing the suspected file with a signature of known crypto-ransomware stored in the signature repository (SR) through SHA-256 hashing. Although this method provides a fast and sure way of detecting known crypto-ransomware, it is very rigid, so even a minor change in the content of a file will result in a mismatch. At the second level, the LA is the trained predictive model using API data from both goodware and crypto-ransomware. This method is considered to be more robust in the detection of both known and unknown crypto-ransomware. Therefore, LA can be expected to detect new or unknown crypto-ransomware. These two detection levels complement each other very well; LA is slow but more robust, while SR is fast but very rigid. The combination of the two detection levels enables PEDA to detect known crypto-ransomware faster, and at the same time robust enough to detect similar behavioural crypto-ransomware with unknown signature.

The signatures of new crypto-ransomware are automatically updated into the SR, which helps to improve detection ability at the first level. However, the performance of the LA in the predictive model, which runs from Step 5 to Step 11 in Fig. 2, must be evaluated to ensure that it can meet its objective. To do so, we applied evaluation metrics to gauge its performance and gain a thorough understanding of the model, which is the focus of this paper. Pseudo code for the PEDA is provided below to provide the reader with a better understanding.

Table 3
Sources of samples.

No.	Source	Amount
1	Old	491
2	VirusTotal	357
3	theZoo	56
4	Goodware	942
	Total	1,846

Table 4
Dataset distribution.

No.	Dataset	Ransomware	Goodware	Total
1	Old	491	942	1,433
2	Full	904	942	1,846
3	PE	205	942	1,147

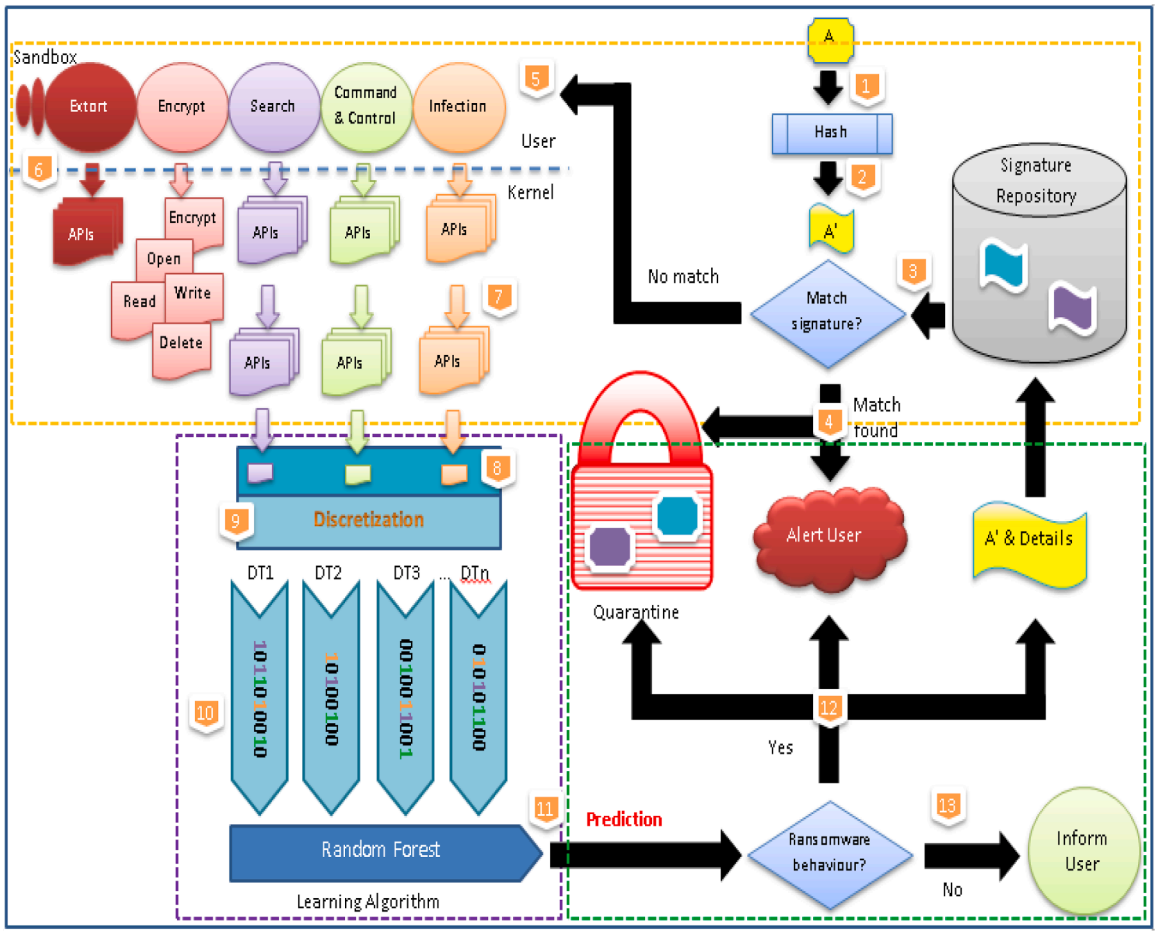

Fig. 3. PEDA process flow.

PEDA Pseudo Code

1. Select file to analyse
2. Produce file signature with SHA-256
3. Match file signature with SR
4. | If match found
5. | | Alert user
6. | | Quarantine the file
7. | If no match found
8. Analyse the file using Cuckoo Sandbox.
9. Generate API through dynamic analysis.
10. Check for encryption API
11. | If found
12. | | Stop
13. | If not found
14. Extract API before encryption API
15. Transform extracted API into data format.
16. Discretize data into discrete data.
17. Test discrete data with Random Forest (RF).
18. RF produced prediction result of the file.
19. | If result is ransomware
20. | | Alert user
21. | | Quarantine the file
22. | | Update the file's signature in SR.
23. | If result is not ransomware
24. | | Inform user
25. End.

3.3. Learning algorithm (LA)

The LA has two phases, the first involving discretization pre-processing to convert the dataset into discrete data. This phase improves the ability of tree-based algorithms to distinguish the data pattern. The second phase involves training and testing the discrete data using the tree-based algorithm known as random forest (RF), using 10-fold cross-validation. RF was selected because it has performed very well in malware detection, as reported in [33]. In addition, it shows better capability than the decision tree in reducing the degree of data bias. We used 10-fold cross-validation to prevent data overfitting by the RF.

3.4. Experiment setup

The experiment was run on a Lenovo Thinkpad x230 equipped with an Intel processor i5-3320M with 8 gigabytes of memory. This laptop was configured to perform a dual boot up of two operating systems, i.e., Ubuntu 18, 04.3 LT and Windows 10 Professional. Cuckoo Sandbox 2.0.7 and its prerequisite programs were installed on the Ubuntu operating system. Samples from Table 3 above were analysed using Cuckoo Sandbox, which then generated a report in JSON format. We wrote a Java program with the capability of extracting API data from the report and converting it into our preferred data format, i.e., CSV format. These steps were taken to create our datasets, as specified in Table 4. Once the datasets were ready, each one was discretized before testing with 10-fold cross-validation of the RF algorithm. The test results provided values for the confusion matrix, from which the conventional and our proposed metrics were calculated.

3.5. Proposed metrics

Metrics are important indicators used to evaluate the performance of a predictive model based on the machine learning approach. Their evaluation results provide a better understanding of the operational strengths and weaknesses of the model. In addition, the proposed metrics described below also provide insights not obtainable by conventional metrics.

3.5.1. Likelihood ratio (LR)

The likelihood ratio (LR) is the likelihood that an API pattern can be identified in ransomware compared to the likelihood that the same API pattern can be found in goodware [34]. There are two types of LR, the positive likelihood ratio (PLR) and the negative likelihood ratio (NLR). In the PLR, a value greater than 10 indicates a strong differentiation between ransomware and goodware, which can be expressed as shown in Eq. (8).

$$PLR = \frac{TPR}{1 - TNR} \tag{8}$$

An NLR value of less than 0.1 also indicates a strong differentiation between goodware and ransomware, which can be expressed as shown in Eq. (9).

$$NLR = \frac{TPR - 1}{TNR} \tag{9}$$

3.5.2. Diagnostic odds ratio (DOR)

The DOR is the ratio between the PLR and NLR, as shown in Eq. (10). This value can range from 0 to infinity, but if it value is greater than 100, this indicates that the predictive model can discriminate between ransomware and goodware [35].

$$DOR = \frac{PLR}{NLR} \tag{10}$$

3.5.3. Youden's index (J)

The J index summarizes the incorrect predictions made by the predictive model, as shown in Eq. (11). When J = 1, this indicates a perfect predictive model with no FPs or FNs [35].

$$J = TPR + TNR - 1 \tag{11}$$

3.5.4. Number needed to diagnose (NND)

The NND determines the number of data required to obtain one correct positive prediction by the predictive model [35]. The smaller is the NND value, the better is the performance of the predictive model, which is expressed as shown in Eq. (12).

$$NND = \frac{1}{[TPR - (1 - TNR)]^{\frac{1}{J}}} \tag{12}$$

3.5.5. Number needed to misdiagnose (NNM)

The NNM determines the number of data required to obtain one incorrect prediction by the predictive model [35]. The higher is the NNM value, the better is the performance of the predictive model, which is expressed as shown in Eq. (13).

$$NNM = \frac{1}{\left[1 - \frac{(TP+TN)}{n}\right]}, \tag{13}$$

where n is the total number of data.

3.5.6. Net benefit (NB)

The NB determines whether the predictive model has provided a correct or incorrect prediction based on a cutoff point in terms of a probability threshold (P), which is expressed as shown in Eq. (14). To visualize how NB varies for different exchange rates, the authors of [36] plotted a graph of NB versus P ranging from 10% to 99% to determine the cutoff point for a positive or correct prediction:

$$NB = \left(\frac{TP}{n} - \frac{FP}{n}\right)\frac{P}{1 - P}, \tag{14}$$

where P is a probability threshold and n is the total number of data.

3.7. Justification for the proposed metrics

The LR utilises the TPR and TNR, which consist of values from all four quadrants of the confusion matrix and ensures that all values are represented and taken into consideration. In addition, the DOR is the ratio of PLR to NLR, which indicates how well the predictive model can differentiate between positive and negative results. This aspect is lacking in conventional metrics. The J index also utilises TPR and TNR to provide a simple indication of the probability of the predictive model producing a correct prediction. NND and NNM provide the range of the number of data required for the predictive model to perform at its optimal level. Any deviation from this range may require the use of additional precautions. The NB, as its name implies, indicates the benefit provided by the predictive model, which can be helpful, especially when comparing its performance with those of other predictive models.

4. Results

The study results can be divided into two groups, as shown in Table 3 and the Data column in Table 4. The first group consists of ransomware samples collected from different sources. This group is used to determine whether the sample has any variations that could affect the performance of the predictive model. The second group consists of different datasets to determine whether the PE dataset, which has comparatively fewer data, can still produce a good predictive model.

Fig. 4 shows that the ransomware from all sources have a PLR greater than 100, and an NLR of zero, which means that the predictive model has good positive and negative likelihood values. In addition, all sources have an infinite DOR value, which cannot be shown in the graph, but which means that the predictive model can distinguish well between ransomware and goodware.

Fig. 5 shows that ransomware from the Old source has a perfect score of 1 for the J index, which indicates that no incorrect predictions were made. However, ransomware from VirusTotal and theZoo had J index values greater than 0.99. This means that data from just one ransomware from all sources was required to make a correct prediction, and 200 or more data were required to make an incorrect prediction. Again, the ransomware from the Old source had an infinite NNM value, which means it made no incorrect predictions.

Fig. 6 shows that the ransomware from all sources had a constant NB

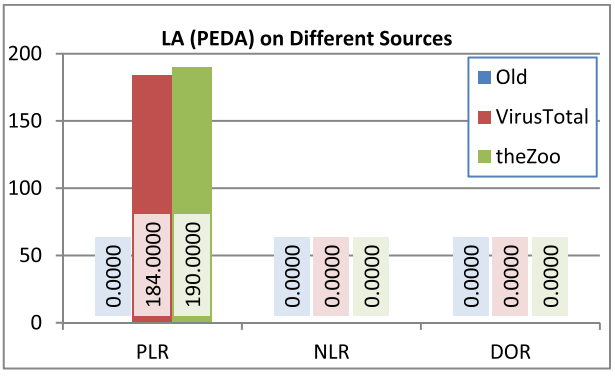

Fig. 4. PLR, NLR, and DOR values obtained for ransomware from different sources.

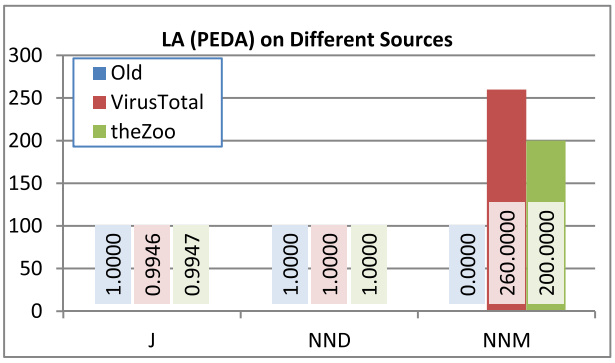

Fig. 5. J Index, NND and NNM for ransomware from different sources.

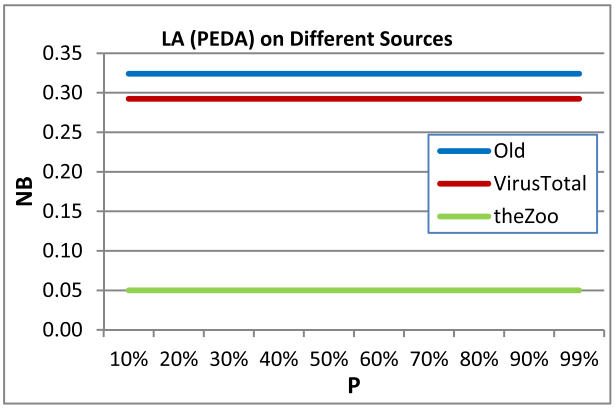

Fig. 6. Decision curve for ransomware from different sources.

for a P ranging between 10% and 99%. This decision curve shows that ransomware from the Old source yielded the highest benefit for analysis, followed closely by ransomware from VirusTotal. However, ransomware from theZoo showed a much lower comparative benefit for analysis.

Fig. 7 shows that all the datasets had a PLR greater than 900, and an NLR less than 0.1, which means that the predictive model has good positive and negative likelihood values. In addition, all the datasets had an infinite DOR value, which cannot be shown in the graph, but which indicates that the predictive model can clearly distinguish between ransomware and goodware.

Fig. 8 shows that all the datasets had a J index value greater than 0.99, which means that the model required just one data from all the datasets to make a correct prediction, and more than 900 data to make an incorrect prediction. This is an optimum operating range for the predictive model.

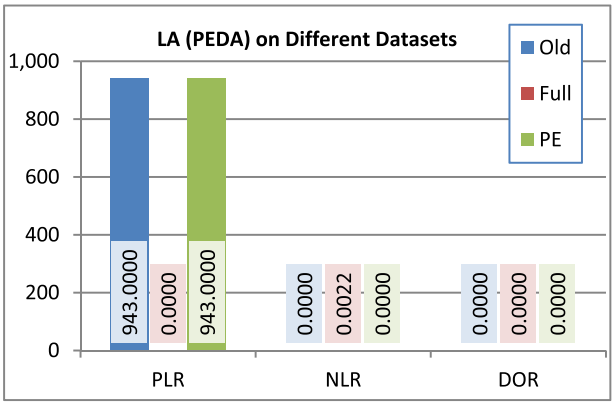

Fig. 7. PLR, NLR and DOR for different datasets.

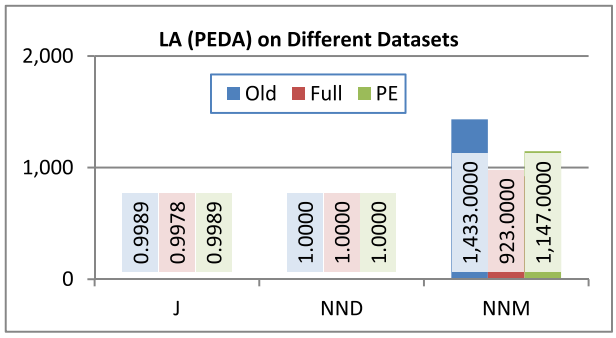

Fig. 8. J Index, NND and NNM for different dataset.

Fig. 9 shows that both the Full and PE datasets have a constant NB for *P* values ranging between 10% and 99%. The Old dataset shows a slowly decreasing NB before P reaches 90%. These decision curves show that the PE dataset had the highest benefit for analysis, with the Full dataset having a lower NB, followed closely by the Old dataset. However, the Old dataset declined quickly once P was greater than 90%, which means that it cannot be used for a high-probability situation.

Fig. 10 shows the FPR and FNR values of the conventional metrics and the NLR of the proposed metric, which indicate good performance of the predictive model when their value is close to zero. The FPR and FNR determine the extent to which the predictive model produced incorrect values, and the NLR determines the likelihood of a positive condition being predicted as negative by the predictive model.

Fig. 11 shows a comparison of the performances of the conventional and proposed metrics. The conventional metrics used include accuracy (Acc), TPR, TNR, precision (Prec), and the F-measure (F). The proposed metrics used include the PLR, NNM, DOR, J Index, and NND. The PLR, NNM, and DOR are shown as continuous lines with values based on the left vertical axis. The DOR values for all the datasets are infinite but are listed as zero in the graph. The PLR value for the Full dataset was also infinite, but is shown as zero. Other metrics with broken lines indicate values for the right vertical axis. Based on the lines shown in the graph, it is clearly difficult to represent all the proposed metrics into one graph, whereas the conventional metrics fit very well in one graph. This shows that the conventional metrics can be used for comparison purposes, but the proposed metrics each provide important insights as individual metrics. In addition, proposed metrics such as the DOR and PLR may have an infinite value, which will be shown as zero on a graph, which could lead to a misunderstanding of the true metric value.

5. Conclusions

The paper showed that it is possible and important to detect crypto-ransomware before encryption happens to prevent the important file from being encrypted, which could result in an irreversible

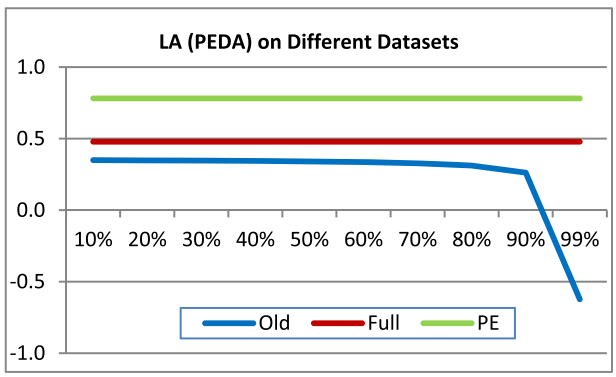

Fig. 9. Decision curves for different datasets.

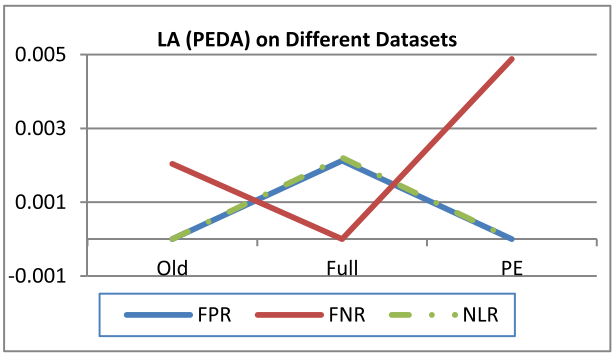

Fig. 10. Comparison of LA results by conventional metrics (FPR and FNR} and proposed metric (NLR) for different datasets.

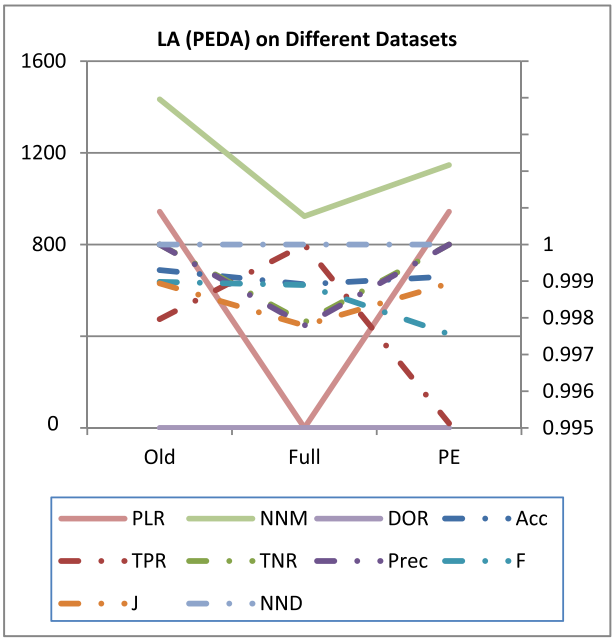

Fig. 11. Comparison of LA results by conventional metrics (accuracy, TPR, TNR, precision and F-measurement) and proposed metrics (PLR, NNM, DOR, J, NND) for different datasets.

consequence. This fulfils the first research contribution. The development of PEDA shows promising results in detecting crypto-ransomware that fulfils the second research contribution. To better evaluate LA performance, in this paper, we proposed six new metrics, including the likelihood ratio, diagnostic odds ratio, Youden's index, number needed to diagnose, number needed to misdiagnose, and net benefit. Use of these metrics addresses the weaknesses of conventional metrics. The likelihood ratio and diagnostic odds ratio can indicate whether the predictive model is able to discriminate between ransomware and goodware. Youden's index can indicate how well the predictive model can produce a correct prediction. The number needed to diagnose and the number needed to misdiagnose indicate the amount of data required for correct and incorrect predictions, respectively. Net benefit indicates how much benefit the predictive model provides by its use. This fulfils the third contribution of this paper.

The results obtained based on the proposed metrics provide new perspectives and insight for the LA. Overall, the metrics indicated that the LA provided exceptionally good performance. The PLR greater than 10, NLR less than 0.1, and DOR greater than 100 confirmed that the LA is able to discriminate between ransomware and goodware. The J index value greater than 0.99 showed that the LA has very low probability of the wrong prediction. The NND of 1 and NNM of more than 1,000 data further proved LA's ability to provide correct predictions. The PE dataset had a constant high net benefit of 0.7817. Although ransomware from theZoo had the least net benefit of 0.0500, due to the smaller sample size, this did not significantly affect the overall dataset generated.

Authors' agreement

This work is the part of PhD program, completed by the scholar, S.H KoK, and supervised by A Azween, and NZJhanjhi.

Declaration of Competing Interest

None.

References

[1] Kok SH, Abdullah A, Jhanjhi NZ, Supramaniam M. Ransomware, threat and detection techniques : a review. Int J Comput Sci Netw Secur 2019;19(2):136–46.
[2] Kong LA, Yeo KN, Ng RX, Kok SH. Ransomware attack and remedial : a survey crypto locky. Int J Innov Res Appl Sci Eng 2020;3(7):490–7.
[3] Kok SH, Abdullah A, Jhanjhi NZ, Supramaniam M. Prevention of crypto-ransomware using a pre-encryption detection algorithm. Computers 2019;8(4):1–15.
[4] Ilangovan R, Chua YK, Kok SH. Ransomware remedial through virtualization. Int J Innov Res Appl Sci Eng 2020;3(7):498–502.
[5] Kok SH, Abdullah A, Jhanjhi NZ, Supramaniam M. A review of intrusion detection system using machine learning approach. Int J Eng Res Technol 2019;12(1):9–16.
[6] Fanshawe TR, Power M, Graziadio S, Ordóñez-Mena JM, Simpson J, Allen J. Interactive visualisation for interpreting diagnostic test accuracy study results. BMJ Evid-Based Med 2018;23(1):13–6.
[7] Gómez-Hernández JA, Álvarez-González L, García-Teodoro P. R-Locker: thwarting ransomware action through a honeyfile-based approach. Comput Secur 2018;73:389–98.
[8] Biomedico CDD, Alberto C. SoLA: social leopard algorithm for intrusion detection honeypot to detect ransomware attacks. IEEE Trans Cogn Dev Syst 2018:16–23. no. Submitted.
[9] Hampton N, Baig Z, Zeadally S. Ransomware behavioural analysis on windows platforms. J Inf Secur Appl 2018;40:44–51.
[10] Zhang H, Xiao X, Mercaldo F, Ni S, Martinelli F, Sangaiah AK. Classification of ransomware families with machine learning based on N-gram of opcodes. Futur Gener Comput Syst 2019;90:211–21.
[11] Monika PZ, Lindskog D. Experimental analysis of ransomware on windows and android platforms: evolution and characterization. Procedia Comput Sci 2016;94:465–72. https://doi.org/10.1016/j.procs.2016.08.072.
[12] Homayoun S, et al. DRTHIS: deep ransomware threat hunting and intelligence system at the fog layer. Futur Gener Comput Syst 2019;90:94–104.
[13] Sgandurra D, Muñoz-González L, Mohsen R, Lupu EC. "Automated dynamic analysis of ransomware: benefits, limitations and use for detection," 2016.
[14] Al-rimy BAS, Maarof MA, Shaid SZM. Ransomware threat success factors, taxonomy, and countermeasures: a survey and research directions. Comput Secur 2018;74:144–66.
[15] Yaqoob I, et al. The rise of ransomware and emerging security challenges in the Internet of Things. Comput Netw 2017;129:444–58.
[16] Tailor JP, Patel AD. A comprehensive survey: ransomware attacks prevention, monitoring and damage control. Int J Res Sci Innov 2017;IV(November):2321–705.
[17] Edgar P, Torres P, Yoo SG. Detecting and neutralizing encrypting ransomware attacks by using machine-learning techniques: a literature review. Int J Appl Eng Res 2017;12(18):7902–11.
[18] Rhode M, Burnap P, Jones K. Early-stage malware prediction using recurrent neural networks. Comput Secur 2018;77(December 2017):578–94.
[19] Stiborek J, Pevný T, Rehák M. Probabilistic analysis of dynamic malware traces. Comput Secur 2018;74:221–39.
[20] Lin CH, Pao HK, Liao JW. Efficient dynamic malware analysis using virtual time control mechanics. Comput Secur 2018;73:359–73.
[21] Pektaş A, Acarman T. Classification of malware families based on runtime behaviors. J Inf Secur Appl 2017;37:91–100.
[22] Chen S, et al. Automated poisoning attacks and defenses in malware detection systems: an adversarial machine learning approach. Comput Secur 2018;73:326–44.
[23] Hamed T, Dara R, Kremer SC. Network intrusion detection system based on recursive feature addition and bigram technique. Comput Secur 2018;73:137–55.
[24] Burnap P, French R, Turner F, Jones K. Malware classification using self organising feature maps and machine activity data. Comput Secur 2018;73:399–410.
[25] Stiborek J, Pevný T, Rehák M. Multiple instance learning for malware classification. Expert Syst Appl 2018;93:346–57.
[26] Kumar A, Kuppusamy KS, Aghila G. A learning model to detect maliciousness of portable executable using integrated feature set. J King Saud Univ-Comput Inf Sci 2016.

[27] Song S, Kim B, Lee S. "The effective ransomware prevention technique using process monitoring on android platform," vol. 2016, 2016.
[28] Kharaz A, Arshad S, Mulliner C, Robertson W, Mulliner C, Robertson W. UNVEIL : a large-scale, automated approach to detecting ransomware. In: USENIX security symposium; 2016. p. 757–72.
[29] Alhawi OMK, Baldwin J, Dehghantanha A. Leveraging machine learning techniques for windows ransomware network traffic detection. Adv Inf Secur 2018; 70:1–11.
[30] Alzahrani A, et al. "RanDroid : structural similarity approach for detecting ransomware applications in android platform," pp. 892–897, 2018.
[31] Cimitile A, Mercaldo F, Nardone V, Santone A, Visaggio CA. Talos: no more ransomware victims with formal methods. Int J Inf Secur 2018;17(6):719–38.
[32] Shaukat SK, Ribeiro V. RansomWall : a layered defense system against cryptographic ransomware attacks using machine learning. In: 10th International Conference on Communication Systems & Networks (COMSNETS); 2018. p. 356–63.
[33] Kok S, Abdullah A, Supramaniam M, Pillai TR, Hashem IAT. A comparison of various machine learning algorithms in a distributed denial of service intrusion. Int. J. Eng. Res. Technol. 2019;12(1):1–7.
[34] Maimó LF, Celdrán AH, Gómez ÁLP, García Clemente FJ, Weimer J, Lee I. Intelligent and dynamic ransomware spread detection and mitigation in integrated clinical environments. Sensors 2019;19(5):1–31.
[35] Bolboacă SD. Medical diagnostic tests: a review of test anatomy, phases, and statistical treatment of data. Comput Math Methods Med 2019;2019:22. https://doi.org/10.1155/2019/1891569. Article ID 1891569.
[36] Vickers AJ, Van Calster B, Steyerberg EW. Net benefit approaches to the evaluation of prediction models, molecular markers, and diagnostic tests. BMJ 2016;352:3–7.