Ransomware Attack Detection by Applying Machine Learning Techniques

Siddharth Routray*[1], Debachudamani Prusti[1] and Santanu Kumar Rath[1]

[1] Department of Computer Science and Engineering,
[1]National Institute of Technology Rourkela, Odisha, India
siddharth.routray@gmail.com, debaprusti@gmail.com,
skrath@nitrkl.ac.in

Abstract. The world faces various perilous threats due to computer security breaches in the present era. It is proliferating at such a fast rate that it hampers the integrity and confidentiality of people as well as organizations resulting in a substantial monetary loss. Among different threats, it has been observed that ransomware is one of its types that results in data loss and makes victims by paying huge ransoms. In this study, a research attempt has been made to detect the attack by applying various machine learning techniques with the dataset. First, the data was trained directly using different machine learning techniques such as k-NN, SVM with different kernel functions (SVM-linear, SVM-Polynomial, SVM-RBF, SVM-Sigmoid), random forest, decision tree, and multilayer perceptron without incorporating any feature selection techniques to detect if the attack is ransomware or benign. Further to optimize the results, feature selection methods based on the filter (Chi-square test, correlation coefficient), wrapper (forward feature selection, backward feature elimination), and embedded methods (LASSO regularization (L1)) are applied to select the prominent features and redundant features are discarded. Finally, all the results obtained from different experiments are analyzed with critical assessment. By investigating the performance measures of various classifiers, it has been observed that significant improvement in the result is being achieved by the machine learning techniques when the feature selection techniques are considered.

Keywords: Classification, Feature elimination, Feature selection, Ransomware

1 Introduction

With the growth of technology and resources, attackers use numerous feasible and intelligent approaches to create malware that serves their purpose. These computer security threats are categorized as computer viruses, spyware, malware, phishing, and many more as shown in Table 1.

Among these threats, ransomware is malware that mainly aims to extortion individuals or organizations. Ransomware enables extortion by planting denial of service to either a system or resources of that system, resulting in the user not accessing the system. In recent times, ransomware is the most used attack vector as it is

irreversible and, unlike other malware, is very difficult to prevent [1]. For example, "SamSam" ransomware in 2018 infected the whole city of Atlanta, the Colorado Department of Transportation, and the Port of San Diego, abruptly terminating services. SamSam ransomware was also used to attack hospitals, municipalities, public institutions, and more than 200 U.S and Canadian companies. SamSam targeted vulnerabilities in File Transfer Protocol (FTP) and Remote Desktop Protocol (RDP) to spread. Another such famous attacks are "WannaCry" ransomware in 2017, which is considered one of the most devastating ransomware attacks in history.

Table 1. Different types of Malwares

Type of Malware	Description
Sniffers	This software keep track of the network traffic analyses them and collects information to initiate malware attack [11].
Spyware	As the name suggests, the spyware collects user information such as passwords, pins, messages without the user knowledge [11].
Adware	This malware tracks users' browsing activity, collects that information, and shows ads on their browsing content and history [11].
Trojan	Trojans disguises itself as a legit application and, upon execution, initiates malicious actions on the user's system [11].
Worms	Upon gaining entry into the user's system, worms replicate themselves exponentially and can cause DDoS attacks and even ransomware attacks [11].
Virus	A virus attaches itself to an application and executes itself upon the execution of that application, causing system harm [11].
Rootkits	A rootkit is an application that enables remote access of the user's system to the attacker [11].
Ransomware	Ransomware is an attack through which an attacker disables a user to access his/her system resources and releases those upon payment of ransom [11].

Ransomware uses asymmetric encryption to encrypt and capture the victim's resources and to decrypt and release the resources, a certain ransom is demanded from the victim. To implement this, a pair of public-private keys is uniquely generated by the attacker for encryption and decryption of resources, whereas the private key required for decryption is provided by the attacker to the victim only after the ransom is paid. After the encryption, ransomware prompts the victim for a ransom and provides

a specific time to make the payment and release the private key on failing which the captive files are destroyed.

According to Chittooparambil et al., none of the existing methods can afford to detect and prevent ransomware attacks [1]. Amongst the difficulties, there is a need to come up with different techniques and methods that is to be used to detect ransomware. This study focuses on finding the most efficient techniques using machine learning and concluding with a pipeline of techniques that gives the best accuracy.

In this paper, we will be providing a brief literature review on this field in section 2 followed by the methodology and techniques applied in section 3. The results and observations are recorded in section 4. Finally, we have concluded the work in section 5 followed by references in section 6.

1.1 Motivation

Ransomware is a type of malware that is hard to detect because of its unique attack style and unique behavior, making it a challenge. Ransoms, i.e., the monetary transaction, are involved in ransomware, making it more interesting. Ransomware data are primarily behavioral and exciting to study and analyses.

2 Literature Review

Ransomware attacks date to 1989 and have gained popularity due to their unique and robust attack style. The first-ever documented ransomware attack is "AIDS Trojan," a PC Cyborg virus spread via floppy disks. Moreover, the ransoms were collected via posts. Ransomware attacks slowly gained more popularity since the 2000s. Furthermore, these attacks got more violent and preferred after the introduction of Bitcoins in 2010.

Eduardo et al. [4] focused on detecting cryptographic ransomware, where they analyzed 63 ransomware samples from 52 families to extract the characteristic steps taken by ransomware. They compared the different approaches and classified the algorithms based on the input data from ransomware actions and the decision procedures to distinguish between benign or malign applications.

P. Zavarsky et al. [7] went on to do an analysis of ransomware on windows platforms and android platforms. For this, they used a dataset of 25 significant ransomware families. 90% of the samples are from Virus Total, 8% are from public malware repositories, and the rest are collected by manually browsing through security forums. Thus, they also found that Windows 10 are pretty effective against ransomware attacks.

Takeuchi et al. [4] detected ransomware using a support vector machine classifier. For the study, they used 276 ransomwares and 312 goodware files. And then extracted a specific ransomware feature known as Application Programming Interface (API). The vector representations of the API call logs resulting from ransomware executions are used as training examples for an SVM. And then, they studied the behavior of the API using Cuckoo Sandbox. They achieved an accuracy of 97.48% by using SVM.

Ban Mohammed Khammas [5] adopted a byte-level static analysis method for detecting ransomware where they achieved high accuracy by using a random forest classifier. Their study has tested different sizes of trees and seeds ranging from 10-1000

and 1-1000, respectively. They also concluded in their study that tree size of 100 with a seed size of 1 achieved an accuracy of 97.74% and a high ROC of about 99.6%. The dataset used by Ban Mohammed Khammas [4] consists of 1680 executable files, including 840 ransomware executable of different families and 840 goodware files.

Li Chen et al. [5] developed a ransomware simulation program to demonstrate how to generate malicious I/O operations on the generated feature sequences (GAN). In this case study, they propose to use GAN to automatically produce dynamic features that exhibit generalized malicious behaviors that can reduce the efficacy of black-box ransomware classifiers.

This section proposes an incremental ransomware detection model using various machine learning techniques. First, we present the architecture of the detection model in Figure 1. The goal is to train the model by feeding the dataset with no feature selection techniques applied and then further optimizing it by applying feature selection techniques on the dataset and improving the classification's accuracy.

3 Methodology

This section proposes an incremental ransomware detection model using various machine learning techniques. First, we present the architecture of the detection model in Figure 1. It comprises four stages: Pre-processing the raw dataset, applying the feature selection technique on it, splitting the data into training and testing data, and then finally training the data using some machine learning classifier models. The goal is to train the model by feeding the dataset with no feature selection techniques applied and then further optimizing it by applying feature selection techniques on the dataset and improving the classification's accuracy. For every feature selection technique, we feed the final features to different classifier models, and finally, we achieve a pipeline of technique and model that gives the best accuracy, as shown in Fig. 1.

3.1 Dataset

The dataset consists of historical data records of data breaches and ransomware attacks over 15 years from 2004 to 2020. The dataset is obtained from The University of Queensland repository [10]. The dataset contains 43 features that define the characteristics, behavior, effect, nature, and aftermath. The feature value in this dataset is in a textual form which further needs to be converted to numerical data in further stages to be processed. The dataset consists of 64 ransomware records, and the remaining are benign records. This study has split the dataset into 70% and 30% training and testing sets, respectively.

3.2 Pre-Processing

The dataset needs to be pre-processed to be used in further phases, which means that the dataset needs to be cleaned for all null, NAN values, and then the features with most NAN values are also removed from the dataset. The dataset we have is in textual form, which cannot be used, so we need to convert the whole textual dataset into numerical

data. For that, we picked every feature and then mapped the unique values of the feature with a unique number. The exact process is repeated and hence making the whole dataset numerical.

3.3 Feature Selection

We take the pre-processed dataset, and before training our model, we first have to check if we can reduce the features and remove obsolete and unnecessary features [8]. Then, we need to apply some feature selection techniques and determine which technique we get a better result based on the result. Then, we take the most efficient feature selection technique to further stages as discussed below.

Filter method. Filter methods are used to eliminate the feature based on their relevancy [8]. This method calculates the intrinsic property for every feature in the dataset via univariate statistics. Therefore, they are cheaper to implement than wrapper methods while dealing with high-dimensional data. The techniques used from filter methods are discussed below.

Chi-Square test. The Chi-square test is best used for categorical values. In this method, two hypotheses are considered i.e., null hypothesis and alternative hypothesis [8]. Then we calculate the chi-square value of every attribute and check if the calculated chi-square distribution is less than 5%. Then, we reject the null hypothesis or accept the null hypothesis. Likewise, we finally get a set of best features at the end to work with.
Correlation Coefficient. This method is used to check the linear relationship of attributes with the target attribute. It picks every attribute from the dataset and checks the correlation between that attribute and the target attribute [8]. If it finds the correlation is very high, we consider that attribute relevant and keep that feature in the dataset.

Wrapper method. It is more complex and expensive than the filter method [8]. In this method, we take every combination of features to create a subset for every combination. Then we train those features using a model, and likewise, we create models for every such possible subset, and finally, we select the features of that subset with the best model results. As the name suggests, the wrapper method wraps the features tests and gives an efficient set of features.

Forward Selection. It starts by creating a subset of features by first selecting the best performing attribute with the target attribute, and then we add the following best-performing attribute until we find the set of best-performing attributes that works best against the target attributes [8].

Backward Elimination. It works exactly the opposite of the forward selection, where we start with training all the attributes against the target attribute and keep on eliminating the best performing attribute until we get the set of best-performing attributes [8].

Embedded method. It works the same as a wrapper but with less computational cost [8]. It is also iterative and checks the usefulness of attributes.

LASSO regularization. While computing the best performing features, LASSO regularization adds a penalty to the model's different attributes that reduce model's

freedom and hence help avoid over-fitting [8]. This process helps reduce computational costs. L1 here has the property to reduce coefficients to zero and remove those features. By continuously removing the features, the best subset of features is obtained.

3.4 SMOTE

Synthetic Minority Oversampling Technique or SMOTE is used for imbalanced datasets to generate synthetic data for minority classes by choosing the minority class as one of k nearest neighbors and balancing the dataset [16]. Since we have an unbalanced dataset in our study, we have used SMOTE to balance the dataset.

3.5 Model Training

Different classifiers such as k nearest neighbor, support vector machine (linear, polynomial, sigmoid, radial basis function), decision tree, random forest, and multi-layer perceptron are used to train the model without the feature selection techniques and then after the feature selection techniques.

k-Nearest Neighbor. It is a machine learning algorithm technique. It is a supervised learning technique. KNN considers various data and categorizes them into specific groups based on their similarities. Upon receiving a new data point, it classifies its group by considering the most k data points near to it [9].

Support vector machine. SVM are supervised learning algorithms suitable for solving classification problems [13] [15]. SVM generates a hyperplane by choosing the extreme points, which then segregates the high dimensional space into different classes, and upon detecting a new data point, it puts that data point into its desirable class [12]. The kernel functions of SVM are discussed below.

Linear kernel. The bare and fast-performing single-dimensional kernel is suitable for classification problems [14].

Polynomial kernel. It is a directional kernel with one or more dimensions. As a result, it is less efficient and accurate than other kernels [14].

Radial basis function. It is more suitable for non-linear problems and when there is no prior knowledge of the data. RBF is the preferred kernel in SVM due to its proper data separation and accuracy [14].

Sigmoid kernel. It is similar to a two-layer perceptron model of a neural network. Moreover, hence is preferred for neural network problems [14].

Decision tree. It is a tree consisting of nodes and branches where each node represents features in an instance that needs to be classified, and branches here represent different node values, and any branch taken is a decision made which leads to the leaf node, which is the outcome of the tree [12] [15].

Random forest. Is also known as decision trees are a collection of many decision trees that overcomes the decision tree limit that is overtraining the model and the instability. Random forest uses different training data for different trees, which reduces the

overfitting problem. However, each tree in a random forest is generated independently and hence takes more complexity to work with random forest [13].

Multi-layer perceptron. It consists of a network of neurons or nodes connected by synapses, and each node and synapse have some weight associated with it. There is one input layer, one output layer, and many hidden layers. MLP works by initially doing an AND operation between the input and weight of the synapses, which generates value at the hidden layers. The hidden layers keep pushing the value after utilizing the activation function at each layer until it generates the output at the output layer. MLP yields outstanding results for classification problems [12].

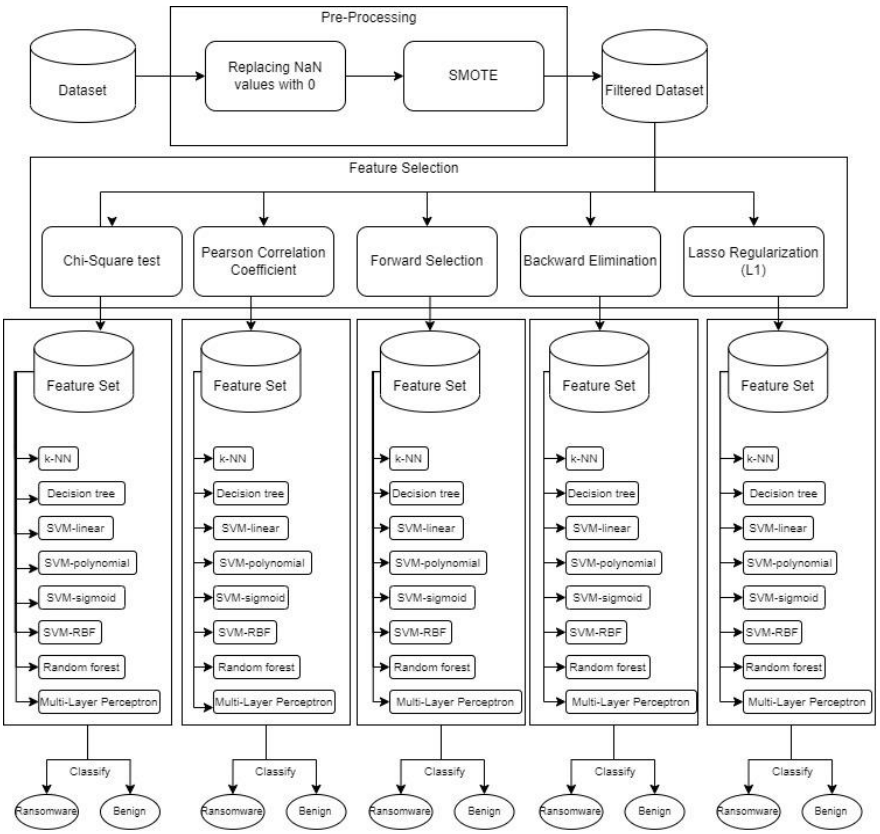

Fig. 1. Proposed Architectural approach with different classification techniques

4 Experimental results

The above-discussed feature selection techniques and classifiers are used, and the observed results are recorded and contrasted. For example, in the following results, the

tangible result is classified into 1 or 0, where 1 represents "ransomware" and 0 means "benign".

4.1 Without any feature selection techniques

Initially, no feature selection techniques were used upon the dataset, and different classifiers were used to check the model's performance. The performance of different classifiers is contrasted in Table 2. Without using any feature selection technique, the decision tree performs better than other classifiers.

Table 2. Contrasting performance of different classifiers in percentage when no feature selection techniques are used.

Classifiers	Accuracy	Precision	Recall	f1-score
k-NN	97.56	20.00	25.00	22.22
SVM-Linear	96.86	22.22	50.00	30.77
SVM-Polynomial	96.52	12.50	25.00	16.67
SVM-Sigmoid	90.24	10.00	75.00	17.65
SVM-RBF	97.21	30.00	75.00	42.86
Decision tree	97.91	75.00	37.50	50.00
Random forest	97.56	20.00	25.00	22.22
Multi-layer perceptron	97.56	20.00	25.00	22.22

4.2 Chi-Square test

To further optimize the results chi-square test is used on the dataset, after which we got 31 features to train further, and then the performance from the different models is contrasted as shown in Table 3. After applying the chi-square test on the dataset and then passing through various classifiers, it has been observed that MLP gave the best performance as compared to other techniques.

Table 3. Contrasting performance of different classifiers in percentage when chi-square test is used.

Classifiers	Accuracy	Precision	Recall	f1-score
k-NN	96.80	25.00	11.11	15.38
SVM-Linear	97.09	50.00	20.00	28.57
SVM-Polynomial	96.22	25.00	9.09	13.33
SVM-Sigmoid	85.47	50.00	4.00	17.41
SVM-RBF	96.51	50.00	16.67	25.00
Decision tree	97.09	12.50	25.00	16.67
Random forest	97.38	14.29	25.00	18.18
Multi-layer perceptron	97.67	16.67	25.00	20.00

4.3 Pearson correlation coefficient

After being applied to the dataset, the Pearson correlation coefficient gave six features to be trained further, and the resulting performance of the classifiers is presented in Table 4. Pearson correlation coefficient gives poor results on the model, and it is found that the k-NN classifier gives the best result here.

Table 4. Contrasting performance of different classifiers in percentage when Pearson correlation coefficient is used.

Classifiers	Accuracy	Precision	Recall	f1-score
k-NN	97.97	33.45	30.91	28.82
SVM-Linear	85.52	3.21	71.43	6.13
SVM-Polynomial	79.34	3.21	71.43	6.13
SVM-Sigmoid	84.01	2.17	71.43	4.22
SVM-RBF	81.23	3.18	71.43	6.10
Decision tree	76.58	3.21	71.43	6.13
Random forest	79.81	3.23	71.43	6.17
Multi-layer perceptron	75.67	3.23	71.43	6.17

4.4 Forward selection

Forward selection is applied with the desired output of a different range of features, and it is found that the best results are obtained when the "k_features" is set for 15 features set. The resultant performance of different classifiers is listed in Table 5.

Table 5. Contrasting performance of different classifiers in percentage when forward selection is used.

Classifiers	Accuracy	Precision	Recall	f1-score
k-NN	97.67	40.00	28.57	33.33
SVM-Linear	97.38	37.50	42.86	40.00
SVM-Polynomial	97.09	28.57	28.57	28.57
SVM-Sigmoid	78.78	7.69	85.71	14.12
SVM-RBF	96.51	27.27	42.86	33.33
Decision tree	97.67	40.00	28.57	33.33
Random forest	97.97	50.00	28.57	36.36
Multi-layer perceptron	97.38	33.33	28.57	30.77

Forward selection technique, when used for feature selection and then training the model with different classifiers, is found that random forest gives the best performance result.

4.5 Backward elimination

The backward elimination technique is used with an output of 20 feature sets after observing the output of various feature sets. The result of different classifiers is shown in Table 6.

Table 6. Contrasting performance of different classifiers in percentage when backward elimination is used.

Classifiers	Accuracy	Precision	Recall	f1-score
k-NN	97.38	33.33	28.57	30.77
SVM-Linear	96.80	25.00	28.57	26.67
SVM-Polynomial	96.80	25.00	28.57	26.67
SVM-Sigmoid	87.79	12.77	85.71	22.22
SVM-RBF	96.22	25.00	42.86	31.58
Decision tree	97.38	33.33	28.57	30.77
Random forest	97.97	50.00	28.57	36.36
Multi-layer perceptron	97.67	40.00	28.57	33.33

Backward elimination technique, when used for feature selection and then training the model with different classifiers, is found that random forest gives the best performance result.

4.6 Lasso regularization (L1)

Lasso regularization (L1) technique is used with an output of 19 feature sets after observing the performance of various feature sets. The performance measures of different classifiers are shown in Table 7.

Table 7. Contrasting performance of different classifiers in percentage when lasso regularization (L1) is used.

Classifiers	Accuracy	Precision	Recall	f1-score
k-NN	97.38	37.50	42.86	40.00
SVM-Linear	97.67	42.86	42.86	42.86
SVM-Polynomial	97.38	33.33	28.57	30.77
SVM-Sigmoid	74.71	6.52	85.71	12.12
SVM-RBF	95.93	29.41	71.43	41.67
Decision tree	97.67	40.00	28.57	33.33
Random forest	98.26	60.00	42.86	50.00
Multilayer perceptron	96.80	30.00	42.86	35.29

Lasso regularization (L1) technique, when used for feature selection and then training the model with different classifiers, is found that random forest gives the best performance result.

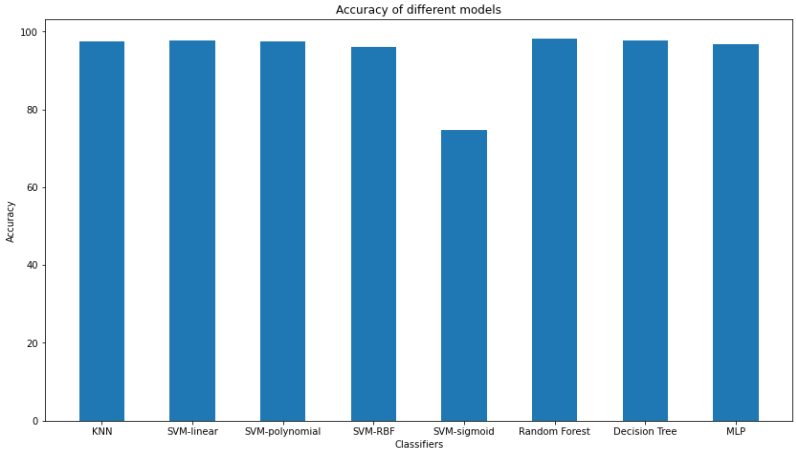

Fig.2. Contrasting different classifier's accuracy after lasso regularization (L1) is applied.

5 Conclusion

In this study, it is intended to find the best-performing model that classifies an attack as ransomware or benign. The dataset is pre-processed and trained directly with different classification techniques such as k-NN, SVM, decision tree, random forest, and MLP to achieve the research objective. Feature selection techniques have been applied to increase the performance further and optimize the results. The feature selection techniques applied with the classification techniques are Chi-square test, Pearson correlation coefficient, forward selection, backward elimination, LASSO regularization (L1). For all the techniques, critical assessment has been carried out to evaluate the performance measures. It has been observed that the incorporation of feature selection techniques has improved the model's performance significantly. Random forest gives the best performance with the LASSO regularization (L1) feature selection technique with a predictive accuracy percentage of 98.26%. Also due to the skewness in the data of the dataset, the important features are considered from the dataset, which improves the predictive accuracy.

This study provides a base model for the classification of ransomware attacks. The model can be further optimized by using different optimization techniques and a combination of different feature selection techniques and classifiers. The future work of this study has a considerable boundary of research to improve and optimize the model and improve the performance.

6 References

1. Chittooparambil, Helen Jose, Bharanidharan Shanmugam, Sami Azam, Krishnan Kannoorpatti, Mirjam Jonkman, and Ganthan Narayana Samy. "A Review of ransomware families and detection methods." In *International Conference of Reliable Information and Communication Technology*, Springer, Cham, pp. 588-597, 2018.

2. Takeuchi, Yuki, Kazuya Sakai, and Satoshi Fukumoto., "Detecting ransomware using support vector machines." In *Proceedings of the 47th International Conference on Parallel Processing Companion*, pp. 1-6. 2018.
3. Ban Mohammed Khammas, "Ransomware Detection using Random Forest technique," In *ICT Express*, vol. 6, pp. 325-331, 2020.
4. Eduardo Berrueta, Daniel Morato, Eduardo Magaña, Mikel Izal, "A Survey on Detection Techniques for Cryptographic Ransomware," *IEEE Access*, vol. 7, pp.144925-144944, 2019.
5. Li Chen, Chih-Yuan Yang, Anindya Paul, Ravi Sahita," Towards Resilient Machine Learning For Ransomware Detection," arXiv: 1812.09400v2 [cs.LG], 2018.
6. P. Zavarsky and D. Lindskog, ‘‘Experimental analysis of ransomware on Windows and Android platforms: Evolution and characterization,’’ *Proc. Comput. Sci*., vol. 94, pp. 465-472, 2016.
7. Mercaldo, Francesco, Vittoria Nardone, Antonella Santone, and Corrado Aaron Visaggio., "Ransomware steals your phone. formal methods rescue it." In *International Conference on Formal Techniques for Distributed Objects, Components, and Systems*, Springer, Cham, pp. 212-221, 2016.
8. Visalakshi, S., and V. Radha. "A literature review of feature selection techniques and applications: Review of feature selection in data mining." In *2014 IEEE International Conference on Computational Intelligence and Computing Research*, pp. 1-6. IEEE, 2014.
9. Guo, Gongde, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer. "KNN model-based approach in classification." In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems*," pp. 986-996. Springer, Berlin, Heidelberg, 2003.
10. Tsen, E., Ko, R. KL., & Slapničar, S. (2020). *Dataset of Data Breaches and Ransomware Attacks over 15 Years from 2004 to 2020*. University of Queensland.
11. Zolkipli, Mohamad Fadli, and Aman Jantan. "An approach for malware behavior identification and classification." In *2011 3rd International Conference on Computer Research and Development*, vol. 1, pp. 191-194. IEEE, 2011.
12. Kotsiantis, Sotiris B., I. Zaharakis, and P. Pintelas. "Supervised machine learning: A review of classification techniques." *Emerging artificial intelligence applications in computer engineering* 160, no. 1 (2007): 3-24.
13. West, Jarrod, and Maumita Bhattacharya. "Intelligent financial fraud detection: a comprehensive review." *Computers & security* 57 (2016): 47-66.
14. Patle, Arti, and Deepak Singh Chouhan. "SVM kernel functions for classification." In *2013 International Conference on Advances in Technology and Engineering (ICATE)*, pp. 1-9. IEEE, 2013.
15. Prusti, Debachudamani. "Efficient intrusion detection model using ensemble methods." *Department of Computer Science and Engineering, National Institute of Technology Rourkela, Rourkela, India*, 2015.
16. Han, Hui, Wen-Yuan Wang, and Bing-Huan Mao. "Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning." In *International conference on intelligent computing*, pp. 878-887. Springer, Berlin, Heidelberg, 2005.
17. S. S. Kshatri, D. Singh, B. Narain, S. Bhatia, M. T. Quasim, and G. R. Sinha, “An Empirical Analysis of Machine Learning Algorithms for Crime Prediction Using Stacked Generalization: An Ensemble Approach,” IEEE Access, vol. 9, pp. 67488–67500, 2021, doi: 10.1109/ACCESS.2021.3075140.
18. Singh, D., & Singh, S. (2020). Realising transfer learning through convolutional neural network and support vector machine for mental task classification. Electronics Letters, 56(25), 1375-1378.