Journal of
Data Acquisition and Processing

ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7765983

FIREFLY BASED FEATURE OPTIMIZATION FOR DEEP TRAINING OF RANSOMWARE DETECTION MODEL

Shweta Shrivastava
PhD Scholar, Department of Computer Science Engineering, RNTU Bhopal

Dr. S. Veenadhari
Associate Professor, Department of Computer Science Engineering, RNTU Bhopal

Abstract— Ransomware is becoming an increasing concern in both the private sector and the public sector, because it can cause immediate financial damages or the loss of important data. The severity of ransomware attacks continues to grow from year to year, along with the associated costs, which have now reached the billions of dollars range. Ransomware attacks have become increasingly common, it is now more important than ever to develop anti-ransomware solutions that can protect users from ransomware. A new ransomware defense system, which is presented in this article is named as FCNNRD (Firefly Convolutional Neural Network based Ransomware Detection). In this FCNNRD firefly genetic algorithm was used to filter good set of features that increase learning of machine. Selected features ere further process for the deep feature extraction by applying convolutional and maxpooling operation. Experiment was done on real malware dataset. Results were compared with existing model on different evaluation parameters.

I. INTRODUCTION

Ransomware is categorized as a type of malware, along with other types of malicious software such as worms, viruses, Trojan horses, hacker utilities, and other types of malware. Other types of malware include hacker utilities [1]. All of the malicious programs are created with the intention of causing damage to the machine that has been infected as well as any other networked machines that contain sensitive data. The most common method for websites and spam emails are the most common vectors for the spread of ransomware. The message that the user has won a prize or that they are eligible for some offers is typically included in the affected webpages or emails. The user will then download the affected links by clicking on the link that they have received [2, 3]. The use of a Trojan horse or any other form of malware as a payload is one of the more common methods of transmission.

There are numerous varieties of ransomware, each of which has its own unique effect on the computer. One variety of ransomware causes the applications, such as internet browsers and security software, to stop functioning altogether. The second type of ransomware is known as "crypto locker," and it encrypts the victim's personal files, such as documents and pictures, preventing access to those files unless a ransom is paid in exchange for the decryption key. Removing ransomware is a difficult task for any business, and the need to have a professional company is one that will help to ensure a safe and secure return to normal operations. The ransomware programs CryptoLocker, Cryptowall, and CTB Locker are just a few examples of the particularly malicious varieties [4].

It is a malicious software program, the creator of which is unknown; however, Adam L. Young and Moti Yung were the ones who first proposed the idea of using public key cryptography for data breach attack in the year 1996. Malware typically takes the form of email attachments or website links that trick users into opening it so that it can encrypt all of the data on a computer and prevent access to it [5]. It is demanded that the victim pay a ransom in order for the files to be decrypted, and if the victim does not comply, the files will be discovered.

II. RELATED WORK

In their study on the detection of android ransomware, Ferrante et al. [6] proposed a hybrid approach. The system incorporated both dynamic and static analysis into its functioning. The frequency of opcodes is relied on by the static detection method, in contrast to the dynamic detection method, which takes into account memory usage, system call statistics, CPU usage, and network usage.

Baldwin and Dehghantanha [7] discovered ransomware by employing the technique of static analysis. They extracted the opcode characteristics as features to be used as input to the machine learning technique represented by the SVM classifier. This was done so that the features could be used in the classification process. In this particular piece of work, the WEKA machine learning toolset was utilised. The best accuracy gained was approximately 96.5% from five different families of crypto ransomware.

The authors Maimó et al. [8] were the first to discuss the impact that ransomware has on clinical settings. WannaCry was the very first piece of ransomware to ever specifically target the healthcare sector. As soon as it was discovered, all of the operations of the NHS were halted, and the vast majority of appointments and surgeries were postponed or cancelled. They came up with a method that was based on ML and was compatible with the architecture of the Integrated Clinical Environment (ICE). With this method, they were able to detect the presence of ransomware before it could even begin to spread. Their method was able to identify variations in the traffic on the network that occurred when the ransomware was being executed. After that, these patterns were input into a probabilistic supervised Ransomware classifier, which was used to finally extract complex characteristics of the sample that was being run. The solution that was suggested consisted of four primary components. The first module analysed the patterns of traffic that were generated by a live sample. The subsequent module required human supervision in order to generate an appropriate dataset that would then be fed to machine learning algorithms for the purpose of ransomware detection and classification. The irregular patterns were classified after being found by the third module, which also gave those names. The most recent unit covered risk reduction strategies with the assistance of rule-based machine learning models.

An automated early detection tool that has the unique capability of pattern extraction was developed by Chen et al. [9]. [Citation needed] Their tool had the capability of preparing an automated analytic report, as well as capturing new strands and samples as they passed through the sandbox. The report was successful in presenting the most distinctive patterns and behavioural paths that are followed by various families of ransomware. Seven different ransomware families were utilized for testing and research purposes by the authors. The authors were able to determine the efficacy of each of the algorithms that were used for pattern extraction by analyzing the results of the experiments that they had conducted. They used TF-IDF, ET, and LDA to automate the process in order to strip the features of various ransomware families. This was done in order to unsheathe the features. The tool that was developed by the authors is suitable for use in medium to large businesses due to its capacity to easily manage large amounts of log data and its ability to detect ransomware before other industry standard solutions.

Imtiaz et al. [10] took on the challenge of resolving the issue of Android Ransomware by employing an innovative methodology known as DeepAMD. DeepAMD utilized deep artificial neural networks (ANNs) for the purpose of detecting ransomware before it could exploit other applications located on the mobile device.

The EBDM that was suggested by Lee et al. [11] makes use of the entropy as the feature, and it has a high rate of detection. However, our most recent research has shown that certain new ransomware has begun to alter the method of encrypting files. Specifically, these ransomware have begun to switch from using complete file encryption to interlaced content encryption in order to increase the encryption speed. This has led to a reduction in the detection rate of the EBDM.

Arabo et al. [12] have used process behaviour analysis to determine whether or not a particular sample of software contains ransomware. The investigation begins by determining which application programming interfaces (APIs) are used and how many system resources are consumed. The result of the dynamic run is analysed to obtain the feature statistics, and then those statistics are fed into a variety of supervised and unsupervised machine learning models. Additionally, file extensions, API calls, and disc usage are taken into consideration during the analysis. The purpose of this particular analysis is to provide users with a timely warning if the binary that is being considered is likely an executable form of ransomware. Their method, on the other hand, has poor accuracy and does not account for false-positive results.

Using dynamic analysis and hash-based techniques, Faghihi et al. [13] have presented a data-centric detection and mitigation against smartphone crypto-ransomware. They make decisions regarding detection based on the user's data while also considering the data's entropy values. API calls that are related to file operations can be intercepted using techniques known as function hooking. After that, the entropy and structure of the data are analyzed in order to find the ransomware and eliminate it. Their method claims to have a high level of accuracy while also having a low rate of false-positive results; however, it demonstrates a low level of resilience to the obfuscation behavior of ransomware executable.

PROPOSED MODEL

Ransomware detection proposed model was brief in this section of paper. Various steps of dataset pre-processing were shown in fig. 1 and explanation of each was also done. Feature selection was done by firefly genetic algorithm and selected features were used for the training of convolutional neural network.

Data Processing

In this module of work input dataset D_{Ran} is processed to get correct set of features that has high impact on learning and detection [14]. For feature filtration two approaches were used first is cleaning where some of column were manually identify and removed. Other technique is genetic based feature selection.

Genetic Feature Selection

In this module input feature vector were used to cluster into two class selected and unselected. Genetic Data Clustering reduces data retrieval time because clustered data access is more relevant and fast.

The intensity of the light helps a firefly swarm travel to brighter and more attractive spots, which may be translated to an ideal solution in the search space. The algorithm standardizes some of the firefly features [15], which can be summarized as follows:

- Regardless of sex, each firefly can be attracted to another.
- The brightness produced by a firefly is directly proportionate to its attractiveness, and when two fireflies are in close proximity, the firefly with higher brightness attracts the firefly with lesser brightness. If a firefly is unable to find a brighter nearby firefly, it will travel aimlessly.
- The objective function determines the brightness of the firefly in the mathematical model.

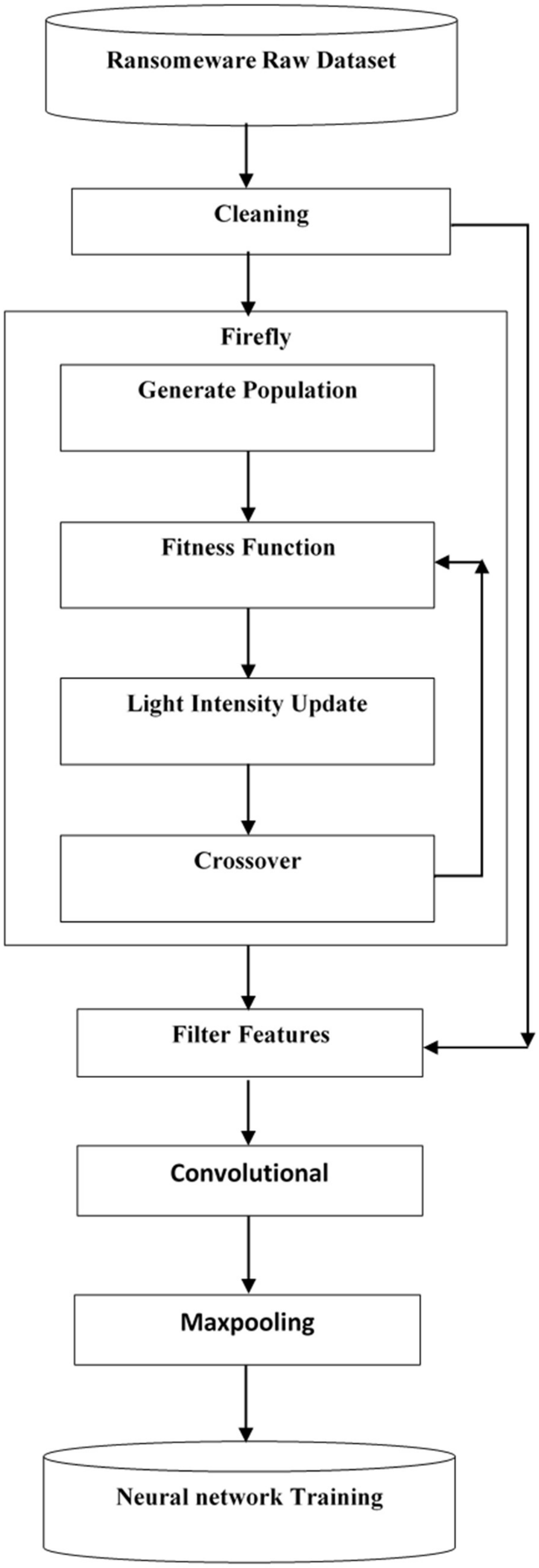

Fig. 1 Block diagram of FCNNRD training model.

Generate Population

Assume some chromosome set that are the combination of different features of session dataset. So chromosome have f number of features Ff={f_1, f_2,…..f_p}. All features in chromosome should have unique set of present features by 1 and absent feature by 0. Now population is set of probable solution hence P_f={Ff_1, Ff_2, Ff_3……..ff_p}.

P_f ←Generate_Population(n, m)

Fitness Function

Firefly searching ability was evaluate by fitness function.

```
Algorithm 1:
Input: CD, T, GM// Cleaned Dataset, Target Output
Output: FV // Fitness Value
Loop 1:m
    Loop 1:f
        S←Selected_Feature(CD) // S: Selected Feature
    EndLoop
    TN←Train(S,T)
    FV[m]←Test(TN, S)
EndLoop
```

Member having highest FV value is consider as producer in the current iteration.

Light Intensity

Calculation of this was done by estimating the total presence of important features available in set [15]. So as per feature presence in dataset intensity value was set.

$$I_p = N_r \times e^{-\tau}$$

Where I_p is intensity of N^{th} node. While τ is constant value range between 0-1 and r is random number vary from 0-1 for each pattern.

Crossover

In this work population P_f chromosome values were modified by best chromosome path. As per fitness value best set was select in the population. Best solution change other set of solutions by replacing node in the path randomly. This crossover generate other set of solution which evaluate and compared with previous fitness value to update the population in the model for next iteration.

Crossover Step

Now check fitness value of this new solution Ffnew, let its fitness value is better as compared to previous one than this Ffnew is insert into population. In similar fashion if Ffnew fitness value is lower than previous solution exist. Hence new updated population is

In similar fashion other set of chromosomes were modified, here it is possible that modification of chromosome were done at more than one place.

Update Population

Once population get new chromosome than it need to filter with best solution sets. Hence fitness value of each were evaluate and the top p solutions from the new set are filter. Once population get update than as per iteration fitness, light intensity and crossover again start. If iteration over than best available solution from the population is consider as final feature set.

Deep Machine Learning

Features were filter as per the firefly algorithm and same were used to train the mathematical model. This work has train the convolutional neural network. It was obtained that use of sigmoid function in training of CNN gives better result.

Convolution

In CNNs, the convolution operation, apply a small convolution mask on the 2D input filtered feature matrix via convolution. There are two main advantages of the convolution operation [16]. (1) The 2D structure of data is preserved. Thus, the convolution-based feature extraction is more effective than the fully connected operation. (2) Because different matrix areas share the same weight parameters, it greatly reduces the number of free parameters in the network. Hence, CNNs are easier to train and less vulnerable to over-fitting.

$C \leftarrow \text{Convolution}(B, s, p, F_c)$--------Eq. 8

Stride is movement speed control variable having integer values. Padding is null row or column add in the block if required. F is filter apply to the B.

Max-pooling

Its basic idea is to enlarge the receptive field by down sampling the matrix feature maps. In CNNs, this down sampling is commonly achieved by maximum pooling or average pooling. After pooling the image feature maps by a factor a, the convolution operation is s times more effective in enlarging the receptive field. To gradually encode high-level image features (from low-level to high-level), the pooling operations usually use s = 2 and the convolution operations and the pooling operations often work together in groups.

$C \leftarrow \text{Maxpooling}\ (C, s, p, F_m)$--------Eq. 8

Fully Convolutional Networks

Convolution operations and the pooling operations allow CNNs to power the spatial information from the input space, the classification formulation prevents the network to utilize the spatial correlation in the output space. To address this problem, fully convolutional networks (FCNs) [16] was proposed. Instead of computing the classes of the feature individually, FCNs directly output the segmentation of the entire feature matrix.

Testing of Trained model

Input testing dataset was processed as done in cleaning step. While features filtration was done by obtained producer from Firefly algorithm during training. Filtered testing dataset was pass in the trained FCNNRD. FCNNRD predicts session class of the input filter dataset.

EXPERIMENT AND RESULTS

Implementation of this model is done on MATLAB 2016 version. Experiment was perform on machine having i3 6th generation processor with an 4GB RAM. In order to do comparison DNAact-Ran model [17] was implement and perform experiment under same environment. Ransomware dataset was taken from [18], having session.

Evaluation Parameters

To test our results, this work uses the following measures Precision, Recall, and F-score. These parameters are dependent on the TP (True Positive), TN True Negative), FP (False Positive), and FN (False Negative).

Precision = TP / (TP+ FP)
Recall = TP / (TP + TN)
F-measure = 2 * Precision * Recall / (Precision + Recall)
Accuracy = (TP+TN)/(TP+TN+FP+FN)
Where
TP : True Positive
TN : True Negative
FP: False Positive
FN: False Negative

Results

Table 1 Ransomware detection models precision value with different testing dataset size.

Dataset Size	DNAact-Ran	FCNNRD
3000	1	1
5000	0.4	0.9995
8000	0.3749	0.6666
10000	0.3	0.6663
20000	0.1999	0.4999

Table 1 shows that proposed FCNNRD model has improved the detection precision value of correct class detection by 40.64% as compared to DNAactRan [17]. Firefly algorithm was used in the model for the detection of good feature set that impact the correct class identification.

Fig,. 2 shows that use of convolutional filter in the training dataset increases the learning of CNN model.

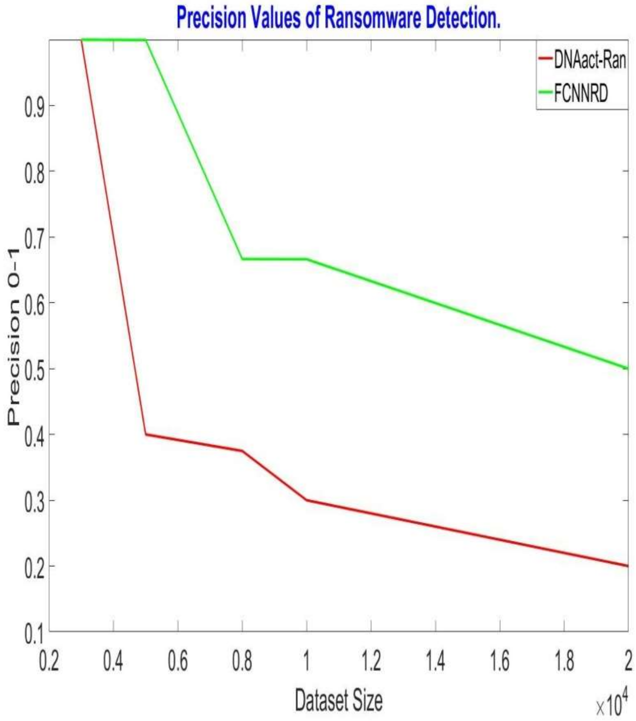

Fig.2 Comparison of different testing dataset size ransomware detection models.

Table 2 Ransomware detection models recall value with different testing dataset size.

Dataset	DNAact-Ran	FCNNRD
3000	0.999	1
5000	0.9995	1
8000	0.9997	0.9995
10000	0.9997	0.881
20000	0.9998	0.5847

Fig. 3 and table 2 shows that ransomware detection at every testing size of FCNNRD model is better as compared to existing model. Use of two genetic algorithms in the model has reduced the feature value as Golf and Cuckoo algorithms make separate feature set.

Table 3 Ransomware detection models f-measure value with different testing dataset size.

Dataset	DNAact-Ran	FCNNRD

3000	0.9995	1
5000	0.5713	0.9997
8000	0.5453	0.7998
10000	0.4615	0.7588
20000	0.3332	0.539

Table 3 shows that proposed FCNNRD model has improved the detection f-measure value of correct class detection by 28.95% as compared to DNAactRan [17]. Firefly algorithm was used in the model for the detection of good feature set that impact the correct class identification. Fig,. 2 shows that use of convolutional filter in the training dataset increases the learning of CNN model.

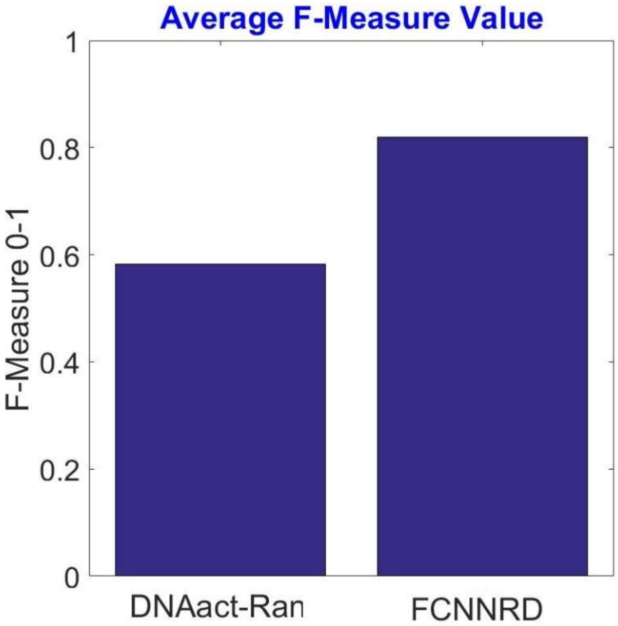

Fig. 4 F-measure based comparison of different testing dataset size ransomware detection models.

Table 4 Ransomware detection models accuracy value with different testing dataset size.

Dataset	DNAact-Ran	FCNNRD
3000	0.9997	1
5000	0.3999	0.9998
8000	0.3748	0.8749

10000	0.3	0.8729
20000	0.199	0.829

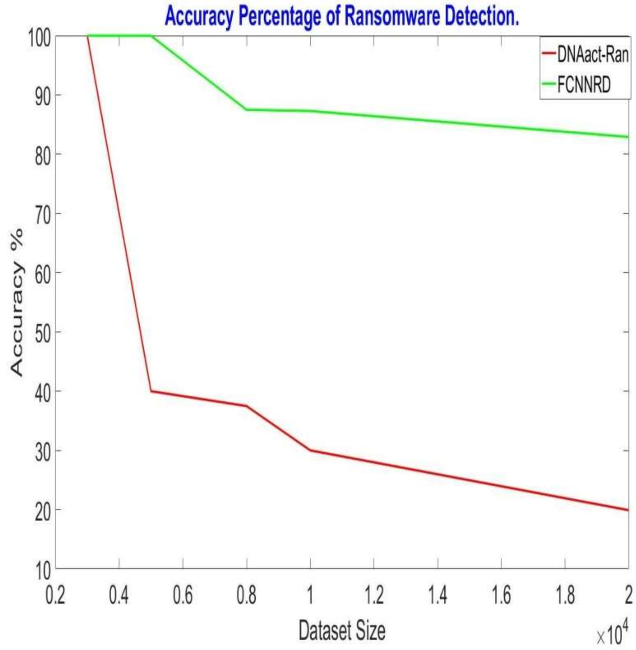

Fig. 5 Accuracy based comparison of different testing dataset size ransomware detection models.

Fig. 5 and table 4 shows that ransomware detection at every testing size of FCNNRD model is better as compared to existing model. Use of two genetic algorithms in the model has reduced the feature value as Golf and Cuckoo algorithms make separate feature set. Further use of firefly algorithm in FCNNRD model has increased the work accuracy value by 50.32%.

CONCLUSIONS

Attack in computer network is serious challenge for the digital and analog communication. Many of measures were adopt for reducing the chance of attack but malicious code, intruder finds new path or vulnerability. This paper has developed a model that works on ransomware detection by learning the network behavior attributes. This paper has proposed a feature reduction model firefly that identify suitable features without prior knowledge. Selected features were used for the training of Deep Neural network CNN. Use of convolutional, maxpooling operation of CNN increase the learning of neural network. Experiment was done on real dataset with different dataset sizes. Result shows that proposed model has increases the comparing parameters values as compared to existing models. It was found that use of firefly algorithm in FCNNRD (Firefly Convolutional Neural Network based Ransomware Detection) model has increased the work precision value by 0.31. In future researchers can work on multinetwork cloud ransomware attack.

REFERENCES

1. B.A.S. Al-rimy, M.A. Maarof and S.Z.M. Shaid, "Ransomware threat success factors taxonomy and countermeasures: A survey and research directions", Computers & Security, vol. 74, pp. 144-166, 2018.
2. K. Burnham, "Emerging Trends in Cybersecurity", Northeastern University Graduate Programs, July 2021,
3. D. Kao and S. Hsiao, “The dynamic analysis of wannacry ransomware,” in 2018 20th International Conference on Advanced Communication Technology (ICACT), 2018, pp. 159–166.
4. M. Humayun, N.Z. Jhanjhi, A. Alsayat and V. Ponnusamy, "Internet of things and Ransomware: Evolution mitigation and prevention. Egyptian Informatics Journal", vol. 22, no. 1, pp. 105-117, 2021.
5. IYaqoob, E. Ahmed, M.H. Rehman, A.I.A. Ahmed, M.A. Al-garadi, M. Imran, et al., "The rise of ransomware and emerging security challenges in the Internet of Things", Computer Networks, vol. 129, pp. 444-458, 2017.
6. Ferrante A., et al. Extinguishing ransomware-a hybrid approach to android ransomware detection International Symposium on Foundations and Practice of Security (2017), pp. 242-258
7. Baldwin J., Dehghantanha A. Leveraging support vector machine for opcode density based detection of crypto-ransomware Cyber Threat Intell. (2018), pp. 107-136.
8. 40. Fernandez Maimo, L.; Huertas Celdran, A.; Perales Gomez, A.L.; Garcia Clemente, F.J.; Weimer, J.; Lee, I. Intelligent and dynamic ransomware spread detection and mitigation in integrated clinical environments. Sensors 2019, 19, 1114.
9. 44. Chen, Q.; Islam, S.R.; Haswell, H.; Bridges, R.A. Automated ransomware behavior analysis: Pattern extraction and early detection. In Proceedings of the International Conference on Science of Cyber Security,Nanjing, China, 9–11 August 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 199–214.
10. 46. Imtiaz, S.I.; ur Rehman, S.; Javed, A.R.; Jalil, Z.; Liu, X.; Alnumay, W.S. DeepAMD: Detection and identification of Android malware using high-efficient Deep Artificial Neural Network. Future Gener. Comput. Syst. 2021, 115, 844–856.
11. K. Lee, S.-Y. Lee and K. Yim, "Machine learning based file entropy analysis for ransomware detection in backup systems", *IEEE Access*, vol. 7, pp. 110205-110215, 2019.
12. Arabo, R. Dijoux, T. Poulain and G. Chevalier, "Detecting ransomware using process behavior analysis", *Procedia Comput. Sci.*, vol. 168, pp. 289-296, Jan. 2020.
13. F. Faghihi and M. Zulkernine, "RansomCare: Data-centric detection and mitigation against smartphone crypto-ransomware", *Comput. Netw.*, vol. 191, May 2021.

14. [12] S. K. Shaukat and V. J. Ribeiro, ‘‘RansomWall: A layered defense system against cryptographic ransomware attacks using machine learning,’’ in Proc. 10th Int. Conf. Commun. Syst. Netw. (COMSNETS), Jan. 2018, pp. 356–363.
15. Jitendra Yadav, Prof. Sumit Sharma, Prof. Pranjali Malviya. "Document Class Identification using Fire-Fly Genetic Algorithm and Normalized Text Features". International Journal of Scientific Research & Engineering Trends, IJSRET.com volume 6 issue 1, 2021.
16. Pratik Shrestha, Aachal Singh, Ishika Sarraf, Riya Garg, Mahesh TR. "Detection and Categorization of Scoliosis Using CNN and SVM Algorithms". IJSET, volume 10 issue 4, 2022.
17. Firoz Khan, Cornelius Ncube, R.Lakshmana Kumar, Seifedine Kadry, Yunyoung Nam. "A Digital DNA Sequencing Engine for Ransomware Detection Using Machine Learning". IEEE Access.
18. https://www.kaggle.com/datasets/nsaravana/malware-detection?select=Malware+dataset.csv