Impact of PCA Feature Extraction Method used in Malware Detection for Security Enhancement

Venkat P. Patil, Hrushikesh Shukla, Sanket Sawant, Zuzer Sakarwala

Abstract: Malware is one of the all told the foremost security threats on the net now a days. Some of the Internet problems like denial of service attacks and spam e-mails have malware threat cause. Computers involved with malware are however networked together for making botnets, and major of threats or attacks are basically launched with the help of these types of malicious and attacker-controlled networks. Downloading files like Executable files like .exe, .bat, .msi etc from sources of untrusted internet probably having an opportunity of getting maliciousness. Further it is seen that these executables are smartly obfuscated with the help of some of the anomalous user for bypassing antivirus stuffs. In this research work , we have proposed an enhanced approach for detecting some of the malicious executables files with the help of analysing the traced Portable Executable (PE) files which are extracted from executable files and use of PCA feature extraction method. The method used in this paper consists of training a supervised binary classifier with the help of these extracted features from the portable executables files from the normal and malicious executables. Considering this approach experimentation has been done on an outsized publicly available dataset and it is seen that over 95% of classification accuracy can be obtained.

Keywords: Malware Analysis ,Machine Learning, , Feature Extraction, PCA feature extraction.

I. INTRODUCTION.

Malware also known as malicious algorithms, which are sent by hackers to infect machines or an entire network of an organization. It exploits device bugs such as a legal program bug related to a browser or web application plugin. Infiltration of malware can have devastating effects like theft of data, extortion, or paralysis of network systems [1,3]. The standard method in malware detection by antivirus programs is scanning a malicious file manually and then generate the signature corresponding to it. Malware can cause multiple damages to a network including data loss, data leaks and hardware failure. For daily notifications, the signatures will be submitted to the client later. Nevertheless, inspecting malicious files manually to acquire the signature may be highly time consuming, boring and mistaken. In fact, to acquire the signature, it requires domain knowledge. Machine Learning methods are used for automating the process of classifying an executable file as harmful or benign in order to overcome the aforementioned limitations. Throughout the study of malicious data, there are primarily two methods, such as static as well as dynamic study. The process of Dynamic solution involves performing a go in a secure environment as a simulator to capture the file's behavioural information and related environmental improvements. Less accurate is the complex analysis. But a completely complex configuration is necessary. For comparison, for the static analysis the dynamic analysis is a little slower. The analysis related to static process, on the other hand, consists of evaluation without it running the components of the executable code. Yet static and dynamic solutions are just as effective. In this paper we are using the Malware Detection for Security Improvement based PCA Feature Extraction Process. We mainly use the corresponding Portable Executable file for review and use of PCA-based feature extraction for checking that the data file is malicious. Windows loader is given the PE file of every executable file. This file includes details such as code length, overlay height, the order of different parts of the file. The latter details allow one to learn how a portable executable (PE) file is executed. This also helps us in viewing malicious images.

This paper contribution is as follow.

a. Retrieving the Portable executable files form the set of given executable files.

b. Performing suitable feature extraction on these portable executable files and then extracts the specified features for the purpose of analysis.

c. Implementing the machine learning classifiers for classifying the malicious and benign executables.

d. Applying PCA feature extraction on this dataset to shortlisted features and achieved a final enhanced accuracy whereas false positive rate was minimized which is tested using precision and recall metrics.

e. Finally testing the obtained results for the performing robust machine learning classifier.

The Rest of this paper is basically briefly summarized as under. We have described the related works in section II and in Section III highlights about the proposed approach followed by experimental set up details and results in Section IV.

Revised Manuscript Received on April 25, 2020.

* Correspondence Author

Venkat P. Patil*, Electronics and Communication Engineering Department, Smt. Indira Gandhi College of Engineering, Navi Mumbai venkat.patil@sigce.edu.in

Hrushikesh Shukla, Computer Engineering Department, Smt. Indira Gandhi College of Engineering, Navi Mumbai. stanhrishi@gmail.com.

Sankat Sawant, Computer Engineering Department, Smt. Indira Gandhi College of Engineering, Navi Mumbai sanketdsawant1998@gmail.com

Zuzer Sakarwala, Computer Engineering Department, Smt. Indira Gandhi College of Engineering, Navi Mumbai. Sakarwalazuzer52@gmail.com

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

Retrieval Number: D8790049420/2020©BEIESP
DOI: 10.35940/ijeat.D8790.049420
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication
© Copyright: All rights reserved.

At the end in Section V ,the concluding remarks and future scope are summarized.

II. RELATED WORK

As mentioned in paper [1] basically an overview on different types of machine learning approaches were summarized for detecting malware . In this paper some of the references are mentioned for exemplifying such methods. As mentioned in paper [2], the concept of boosted decision trees working on the principle of n- grams are found for producing very good results than that of the Naive Bayes classifier and SVM-Support Vector Machines. Author of [3], Shafiq et al., elaborated about employing techniques of data mining on Portable Executable files for determining malicious files. Basically they obtained their dataset from VX heavens and Malfese. In this paper they got accuracy of through their method. Author Shabtai at el. explained about different taxonomies that classified methods of detection required for malicious codes by using Machine Learning algorithms using the approach of extracting static features from executables files [4]. Authors in paper [5] basically used a dataset of related 30,000 samples and then performed operation of representation of byte sequence n-gram while considering a probability to achieve accuracy of 95% and with malicious samples lesser than 20% . In their paper ,they basically made the evaluation of representation of Opcode n-gram and further they claimed that it is possible to get more than 99% accuracy along with dataset of lesser than 15% malicious content which they have found more than their earlier practical experimentation [5]. As mentioned in paper [6] "Hidden Markov Models" are basically used for detecting to test about a given executable program file is malicious or is not or a variant of a previous program file. For reaching a similar requirement , author [7] basically employs principle of "Profile Hidden Markov Models", which they had earlier used with great success for the purpose of sequence analysis related to bioinformatics application . Authors Singhal and Raul as mentioned in their paper [8] summarized an antivirus engine which basically extracts API calls and then applies ML approach for detecting malicious executable files . As explained by author in paper [9] , basically they made use of automatic extraction of association rules on the platform of "Windows API execution sequences" for distinguishing between clean and malware program files. Further use of association rules, but considering honey tokens of known parameters, is mentioned in paper [10]. Authors Vyas at el. Highlighted an approach for detecting portable executable files on network with the help of ML algorithms. In this approach basically they summarized 28 features which were basically extracted from the sources of packing and metadata imported from DLL files. It was found that their system shows an accuracy of 98.7% and also 1.8% false positive rate as mentioned in paper [11]. As demonstrated by authors in paper [12], principle of "Self-Organizing Maps" are used for identifying patterns of behaviour for viruses in Windows executable files.

III. METHODOLOGY

The main aim of this particular section is to switch our earlier the algorithm [13], so on correctly detect malware files and at the same time forcing (as far as possible) a 100% detection rate for one category. The main focus during in this paper is for analysing the suitable features extracted from Portable Executable of an executable by using PCA based enhanced feature extraction for detection of malware for security enhancement. Portable Executable of an executable may be a collecting the information elements which are basically required by windows loader platform . Portable executable file basically contains various basic elements like information overlay number, information sections size and size of code. With help of these Portable executable file, we will get idea of execution of the program.

A. The basic model architecture

The basic model architecture is shown in Figure 1 and 2 . Here we have shown the architecture of two models of our system models. For testing purpose We have considered dataset and then accordingly trained our model. At the end of finishing the training phase , we have used the executable files and then extracted suitable features from it and after that passed it further through the trained model for the purpose of evaluation . It is to be noted that the measure used for the purpose of evaluation of this model is accuracy and also false positive rate. Our proposed model is basically a two class classifier which takes basically input as an executable file and offers "output as a label showing whether a file is malware or not."

B. The System Components used:

The basic system model components for both the models A and B as shown in figure 1 and 2 are described as follow.

1) **Dataset used** : In this paper we have trained our model using a dataset available on Kaggle for free. The dataset is in form of csv file which basically consists of features extracted from Portable Executable files.

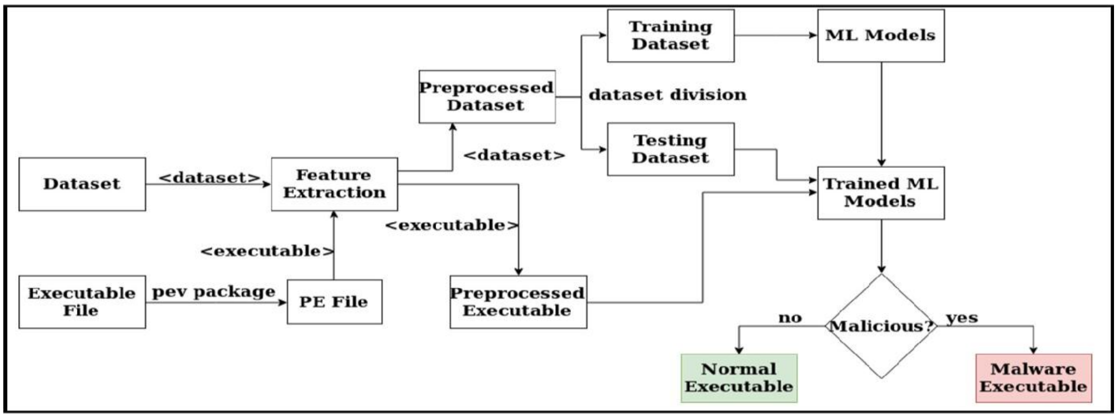

Figure 1: System Architecture -A. [13]

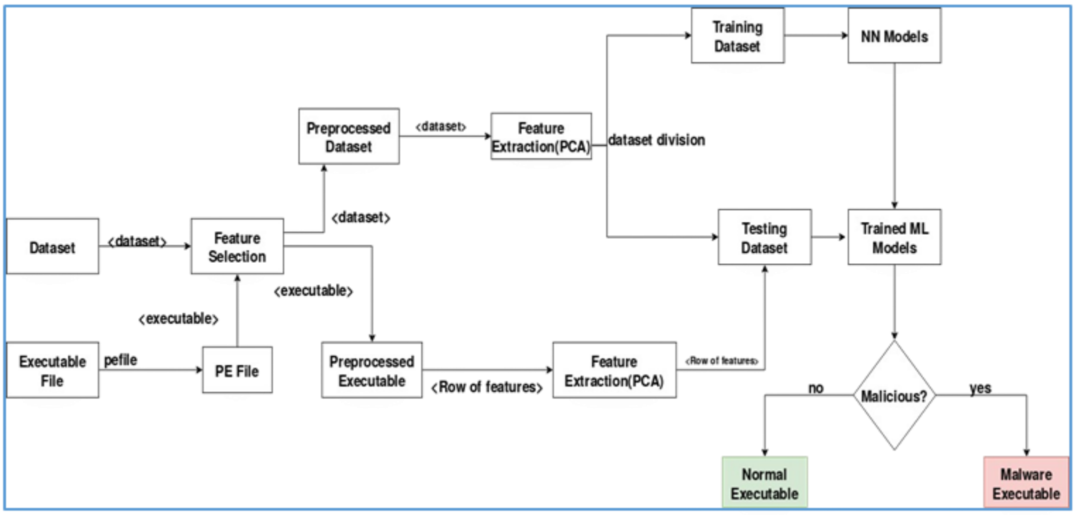

Figure 2: System Architecture -B.

2) Portable Executable Files: Basically this component helps us regarding inputting an executable file in our proposed system. After that It performs the operation of extracting features of the Portable executable file from the input executable files and organizes that of PE file in the text file in a defined format. We can also extract all the required features of PE with help of same text file .These features extracted are then further processed by passing on to the Feature Extraction block which extracts the required or suitable features.

3) Feature Extraction process: Basically the input to this particular block can be a row from dataset under consideration or it can be extracted features from Portable executable file. The purpose of this block is to select the suitable features from the input data under consideration and completes the respective pre-processing operation .

4) Machine Learning Classifiers:

The processed dataset used is further spitted into testing and training sets . We have considered variable size of testing and training data sets for getting different results. We changed the training and testing sizes as per multiple ratios for getting various results. Further various machine learning methods are applied and then compared them as per their performance on testing set accordingly.

5) Trained model: The best performing model is saved and which can be used to make predictions on the testing files. This model will help basically to make predict regarding the given input file is malicious or benign.

C. System components based on Feature Extraction Selection Using PCA:

It is seen that redundant and Irrelevant features can lead to an machine learning classifier to finish training very slowly and thus perform less accurately. In this paper, the PCA approach which is also known as "the eigenvector regression filter or the Karhunen-Loeve transform" is used for feature selection purpose, which basically involves removing one or more of the weakest principal components based on Eigen values and variance, the resulting subset of raw features is sufficient enough for preserving maximum data variance. The modified architecture comprising of PCA based feature extraction is as shown in figure 2.

IV. EXPERIMENTAL SETUP AND RESULTS

we have discussed in this particular section about the different experimental set up and results that were obtained by using different machine learning approaches on same data set for two models A and B as shown in figure 1 and 2.

A. Consideration of Data Description and pre-processing

We downloaded our publicly available dataset from Kaggle. The dataset is in form of csv file, where each row describes information about a particular executable. Each row consists of 75 features like size of overlay, magic number, section entropies, etc. extracted from PE files. it is to be mentioned that the last column of dataset basically represents the label of the file; this label says whether the file is malicious or benign. We selected 51 features out of the total 75 as rest of them contained common or repeated values. As a solution to this problem, we scaled all the values in our dataset between 0 and 1.

B. Basic Experimental consideration and results for model –A

A The Machine learning model in our case takes input as pre-processed features extracted from Portable executed files and gives its prediction regarding whether the input features are from malicious or benign file. The algorithms we considered for classification are Decision tree, Naive Bayes, Random forests, Artificial Neural Networks(ANN) and Logistic Regression. We trained each of these models on various percentages for training and testing. First 33% of total dataset for testing and 67% of the same for training. We then took 10% of total data for testing and 90% of the total dataset for training. As we have 10,000 samples in total, 10% of it (1000 samples) would be sufficient for testing purpose. Fig. 3 and 4 shows the performance of different machine learning algorithms in terms of accuracy before and after scaling of data. Figure 3 shows accuracy with testing size is 33% and training size is 67% and Figure 4 shows accuracy with testing size is 33% and training size is 67%.

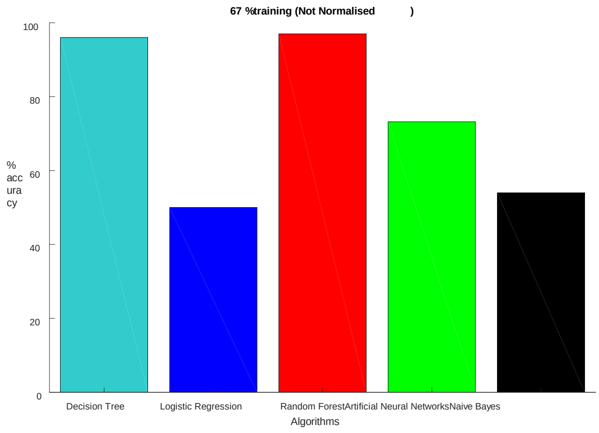

Figure 3: Results with training size is 67% and test size is 33%.

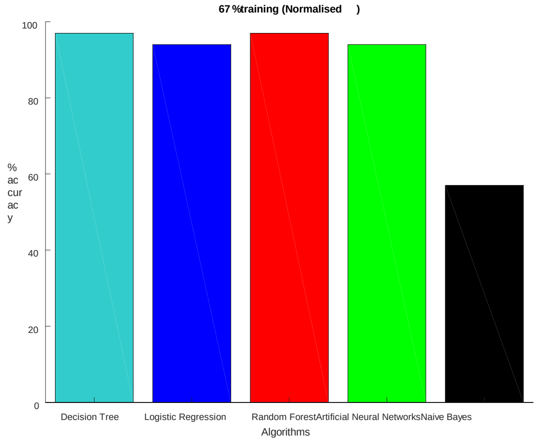

Figure 4: Results with training size is 67% and test size is 33%.

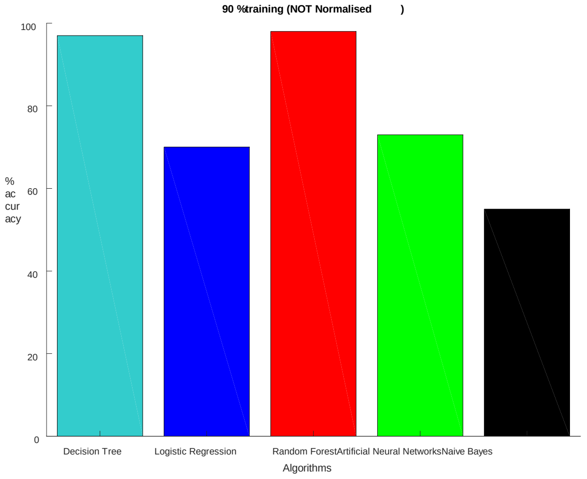

Figure 5: Results with training size is 90% and test size is 10%.

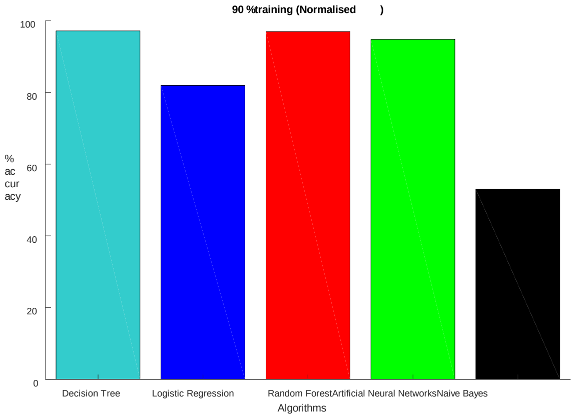

Figure 6: Results with training size is 90% and test size is 10%.

Figure: 5 and 6 shows the performance of the used method or algorithms before and after pre-processing the data by considering the size of testing is fixed to 10% and at the same time considering the size of training is fixed to 90%. Figure 5 shows accuracy with testing size is 10% and training size is 90%. Whereas Figure 6 shows accuracy with testing size is 10% and training size is 90%.

Retrieval Number: D8790049420/2020©BEIESP
DOI: 10.35940/ijeat.D8790.049420
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication
© Copyright: All rights reserved.

Table 1: Results without and with PCA based feature extraction.

Data	76 columns				51 columns				30 columns	
Metric	(without feature selection)				(with feature selection)				(feature extraction by PCA)	
processed	Raw data		Processed data		Raw data		Processed data		Processed data	
Confusion matrix	TP=0 , FN=501	FP=0 ,TN=502	TP=400 ,FN=93	FP=90 ,TN=420	TP=450 ,FN=83	FP=80 ,TN=390.	TP=470 ,FN=32.	FP=41 ,TN=460	TP=486, FN=15	FP=41 , TN=461
Accuracy	0.5		0.817		0.837		0.927		0.944	
False positive rate	0		0.17		0.17		0.081		0.081	
precision	0		0.816		0.849		0.919		0.922	
recall	0		0.811		0.844		0.936		0.97	

C. Experimental Setup and Results for Model B:

Results Using PCA Based Feature Extraction is shown in Table 1 which shows Results without and with PCA based feature extraction. This consists of 3 different considerations as shown in table 1 In first one we kept all the 76 features that we had initially. When the data was un processed we can see the accuracy was quite low i.e. 50%. Even We had zero precision and recall values. Whereas after processing the data you can see that we got 81% 0f accuracy and precision and recall of 81.6 and 81.1 percent respectively. In the second case we had only 51 of the total 76 features. These 51 features were selected manually. We did this feature extraction by removing unwanted features for example: machine identifier: It tells us from which machine, the file was collected, it's a un necessary feature so we remove it. Once we start working on this 51 features without pre-processing we got accuracy of 83% with false positive rate of 0.170 and precision and recall of 84.9 and 84.4%. On the other hand, after pre-processing this data we got a better accuracy of 92.7% and false positive rate of 8.1%. Lastly in the third case, we then applied PCA feature extraction on this dataset to shortlist 30 features and achieved a final accuracy of 94.4% whereas false positive rate was of 8.1% along with precision and recall of 92.2% and 97% each.

V. CONCLUSION

Malware is considered as a serious security threats on the Internet today. It is found that most problems related to internet like denial of service attacks and spam e-mails have basically malware as their threat . In this research paper, it has been demonstrated that the design of a binary classifier having its robustness , which basically performs the operation of classifying the files into malicious and further benign with higher accuracy. The model demonstrated in this paper enhances the security mechanism with the help of PCA based feature extraction and detects regarding the file is malware or not. By using basic model working on the 51 features without pre-processing we got accuracy of 83% with false positive rate of 0.170 and precision and recall of 84.9 and 84.4%. On the other hand, after pre-processing this data we got a better accuracy of 92.7% and false positive rate of 8.1%. We then applied PCA feature extraction on this dataset to shortlist 30 features and achieved a final accuracy of 94.4% whereas false positive rate was of 8.1% along with precision and recall of 92.2% and 97% each. Thus we get enhanced results with the proposed model based on PCA feature extraction. The proposed further work of this paper might elaborate basic actions to be taken while the file is detected to be malicious by using further modified algorithm.

REFERENCES

1. P. K. Chan and R. Lippmann, "Machine learning for computer security," Journal of Machine Learning Research, vol. 6, pp. 2669–2672, 2006.
2. J. Z. Kolter and M. A. Maloof, "Learning to detect and classify malicious executables in the wild," Journal of Machine Learning Research, vol. 7, pp. 2721–2744, December 2006, special Issue on Machine Learning in Computer Security.
3. M. Z. Shafiq, S. M. Tabish, F. Mirza, and M. Farooq, "Pe-miner: Mining structural information to detect malicious executables in realtime," in International Workshop on Recent Advances in Intrusion Detection. Springer, 2009, pp. 121–141.
4. A. Shabtai, R. Moskovitch, Y. Elovici, and C. Glezer, "Detection of malicious code by applying machine learning classifiers on static features: A state-of-the-art survey," information security technical report, vol. 14, no. 1, pp. 16–29, 2009.
5. R. Moskovitch, D. Stopel, C. Feher, N. Nissim, and Y. Elovici, "Unknown malcode detection via text categorization and the imbalance problem," in 2008 IEEE International Conference on Intelligence and Security Informatics. IEEE, 2008, pp. 156–161.
6. M. R. Chouchane, A. Walenstein, and A. Lakhotia, "Using Markov Chains to filter machine-morphed variants of malicious programs," in Malicious and Unwanted Software, 2008. MALWARE 2008. 3rd International Conference on, 2008, pp. 77–84.
7. M. Stamp, S. Attaluri, and S. McGhee, "Profile hidden markov models and metamorphic virus detection," Journal in Computer Virology, 2008.
8. P. Singhal and N. Raul, "Malware detection module using machine learning algorithms to assist in centralized security in enterprise networks," arXiv preprint arXiv:1205.3062, 2012.
9. Y. Ye, D. Wang, T. Li, and D. Ye, "Imds: intelligent malware detection system," in KDD, P. Berkhin, R. Caruana, and X. Wu, Eds. ACM, 2007, pp. 1043–1047.
10. M. Chandrasekaran, V. Vidyaraman, and S. J. Upadhyaya, "Spycon: Emulating user activities to detect evasive spyware," in IPCCC. IEEE Computer Society, 2007, pp. 502–509.

11. R. Vyas, X. Luo, N. McFarland, and C. Justice, “Investigation of malicious portable executable file detection on the network using supervised learning techniques,” in 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM). IEEE, 2017, pp. 941–946.
12. I. Yoo, “Visualizing Windows executable viruses using self-organizing maps,” in VizSEC/DMSEC ’04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security. New York, NY, USA: ACM, 2004, pp. 82–89.
13. Hrushikesh Shukla, Sonali Patil, Dewang Solanki, Lucky Singh, Mayank Swarnkar, Hiren Kumar Thakkar. "On the Design of Supervised Binary Classifiers for Malware Detection Using Portable Executable Files", 2019 IEEE 9th International Conference on Advanced Computing (IACC), 2019.

AUTHORS PROFILE

Venkat P Patil is graduated in BE and M. TECH and is working as faculty of engineering and technology working since 30 years under various engineering colleges of university of Mumbai affiliated college**s.** Currently working as Vice-Principal /Associate Professor in Smt. Indira Gandhi engineering college, Navi Mumbai since 25 years. His area of research is image processing and computer vision and computer security. He has published more than 50 research papers and more than 10 patents in the field of Engineering and Technology.

Hrushikesh Shukla is an Under graduate student of BE computer engineering studying in Smt. Indira Gandhi engineering college, Navi Mumbai and his areas of interest are information security, machine learning, deep learning, big data. He has published more than 2 research papers and 1 patent in engineering and technology.

Sanket Sawant is an undergraduate Student of B.E Computer engineering studying in Smt. Indira Gandhi engineering college, Navi Mumbai and his areas of interest are information security, machine learning, deep learning, big data.

Zuzer Sakarwala is an undergraduate Student of B.E Computer engineering studying in Smt. Indira Gandhi engineering college, Navi Mumbai and his areas of interest are information security, machine learning, deep learning, big data.

Retrieval Number: D8790049420/2020©BEIESP
DOI: 10.35940/ijeat.D8790.049420
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication
© Copyright: All rights reserved.