

Ransom Prediction Using ML Algorithm

**Kanakam Maruthi Narasimha Rao¹, Konda Saiteja², Kamini Vasu³,
Raj Kumar P⁴, Dr.K.S.Ramanujam⁵**

^{1, 2, 3}Student, ^{4, 5}Assistant Professor
Dr MGR Educational and Research Institute

Abstract

As wireless communication has advanced, there are several online security risks. The ransomware prediction system assists in identifying system threats and malware detection. In the past, a variety of machine learning (ML) techniques have been used to try to improve the accuracy of the ransom malware system and the outcomes of malware detection. Using the random forest classification technique and principal component analysis (PCA), this research has suggested a method for creating an effective ransomware system. Whereas the random forest will aid in classification, the PCA will assist in organizing the dataset by lowering its dimensionality. According to the results, the suggested method outperforms other methods like SVM, Naïve Bayes, and Decision Trees in terms of accuracy. Performance time (min) is 3.24 minutes, accuracy rate (%) is 96.78%, and error rate (%) is 0.21% according to the results of the suggested approach.

Keywords: Online Security Risks, Ransom Ware, ML, Random Forest, PCA, Outperforms

INTRODUCTION

The development of wireless communications has led to a notable rise in online security risks. Ransomware prediction tools are essential for identifying malicious software and system threats. Several machine learning (ML) approaches have been used in earlier studies to detect ransomware with the goal of increasing detection capabilities and accuracy. Using the Random Forest classification technique and Principal Component Analysis (PCA), this research suggests an effective ransomware prediction system. The Random Forest approach enhances classification performance, while PCA helps structure the dataset by lowering its dimensionality. According to the results, the suggested method performs more accurately and efficiently than other methods like Support Vector Machine (SVM), Naïve Bayes, and Decision Tree. According to the evaluation, the system achieves a runtime of 3.24 minutes, a 96.78% accuracy rate, and a 0.21% error rate.

Any activity that jeopardizes the availability, confidentiality, and integrity of data or computer resources is considered malware. Cybercriminals circumvent authentication or authorization procedures by taking advantage of holes or defects in the computer architecture. Network security is now more important than ever due to the quick growth of network-based services and the increasing demand for safe data transfer. Ransomware prediction systems, which keep an eye on a variety of network activity, are a crucial tool for identifying assaults. These systems need to be able to reduce false positives, train quickly, and identify threats with high accuracy. By identifying dangerous abnormalities, offering insight into odd

activity, alerting administrators, and stopping malicious activity, a ransomware prediction system helps safeguard networks. Ransomware prediction systems come in two main varieties: host-based and network-based. The goal of network-based ransomware detection is to employ network traffic analysis to find unusual and unauthorized activity. However, in order to identify malicious activity within a host, host-based intrusion detection systems (HIDS) keep an eye on and examine system logs, file integrity, and process behavior. Because they guard against system hacks, data breaches, and unwanted access, both kinds are essential to maintaining cybersecurity. Using an effective and precise ransomware prediction system is essential for safeguarding digital assets and preserving network integrity in light of the growing sophistication of cyberthreats.

RELATED WORK

We offer a signature-based intrusion detection system design that includes methods to identify all of the aforementioned threats. According to our test results, the IDS cannot identify any of the nine assaults, and the AP is susceptible to eight of them. The primary drawback of this technique is that none of them can be detected by IDS [1].

Then, utilizing the condensed feature space, we use the following machine learning techniques: Decision Tree (DT), Support Vector Machine (SVM), k-Nearest-Neighbor (kNN), Logistic Regression (LR), and Artificial Neural Network (ANN). Both the binary and multiclass classification configurations were taken into account in our investigations. The findings showed that the XGBoost-based feature selection method enables techniques like the DT to improve the binary classification scheme's test accuracy from 88.13 to 90.85%. One of its drawbacks is that, generally speaking, it needs to be carefully adjusted, particularly when the input dimension exceeds the number of samples [2].

We compared five machine learning algorithms: Logistic Regression, Decision Trees, random forests, XGB, and Artificial Neural Network. We have created a new Artificial Neural Network architecture that surpassed performance compared to the other algorithms obtaining ~99.2 % accuracy. We conducted our experiments on a Linux machine running Ubuntu 21.04 OS with 32 GB RAM and 8 GB NVidia GTX

1080 GPU to use distributed training of our algorithms to obtain results faster. They have obtained an accuracy of only around 81% on the ADFA Intrusion Detection dataset. Anomaly-based detection has its disadvantages [3].

Use the Attack Intention Analysis (AIA) method to forecast the attack Let us begin by discussing this system's structure. While research in the creation of defense systems focuses on the monitoring and analysis of assaults without citing actual remedies to these attacks, the typical intrusion detection system automatically monitors and analyzes attacks that are indicative of the existence of the human element. This process is challenging to manage [4].

The effectiveness of the suggested KDE-HMM method. To improve results, the suggested KDE-HMM technique/method combines the benefits of both statistical and probabilistic features. Experimental validation has confirmed the effectiveness of the suggested KDE-HMM approach, which identifies the aforementioned threats with 98% accuracy. Their approach's main drawback is that it uses a preset threshold to choose the best-ranked attributes. Therefore, in order to detect invasions, the current intrusion detection systems (IDS) that are based on stochastic models, like HMM, only analyze the behavior of the attacker and the initial infection vectors, such as system calls, system activities, and

signatures. Furthermore, attackers can handle and control system calls, thus these parameters are insufficient to detect an intrusion [5].

EXISTING SYSTEM

Several machine learning techniques for the intrusion detection system were examined by Iftikhar Ahmad et al. They contrasted a few methods, including random forest, SVM, and extreme learning machines. According to the authors' findings, the Extreme machine learning approach outperforms all other techniques. Here, B. Riyaz et al. sought to enhance the dataset's quality in order to supply it to the intrusion detection system. To enhance the dataset, they employed a feature selection method based on fuzzy rules. Using the KDD dataset, they demonstrated dynamic increase in the IDS result.

Disadvantages

Numerous harmful actions affect systems that operate over the internet. The main issue in this subject is information violations caused by system intrusions. According to current findings, there might be room for improvement in terms of accuracy, detection rates, and false alarm rates. Other methods, including SVM and Naïve Bayes, can take the place of previously used methods. Additionally, the paper claims that there are ways to improve the dataset. To raise the standard of input for the suggested system.

REQUIREMENT ANALYSIS

System Requirements

Hardware Requirements:

- *System : Pentium IV 2.4 GHz.*
- *Hard Disk : 40 GB.*
- *Floppy Drive: 1.44 Mb.*
- *Monitor : 15 VGA Colour.*
- *Mouse : Logitech.*
- *Ram : 512 Mb*

Software Requirements:

- *Operating system : Windows 7*
- *Coding Language : Python*

PROPOSED SYSTEM

The purpose of the ransom malware prediction system is to enhance the system that is impacted by malware. This system is capable of detecting malware. The suggested system attempts to resolve the issues that still persist from the earlier work. Principal component analysis and random forest are the two techniques that make up the suggested system. Principal component analysis is used to reduce the dataset's dimensions; this technique will increase the dataset's quality because it may contain the right qualities. Following this, the random forest algorithm—which outperforms SVM in terms of both detection rate and false alarm rate—will be used to detect Trojan horses.

Advantages:

Our suggested method has an extremely low error rate of 21%. Additionally, compared to earlier algorithms, the accuracy achieved is far higher. Additionally, compared to other algorithms, the performance takes less time.

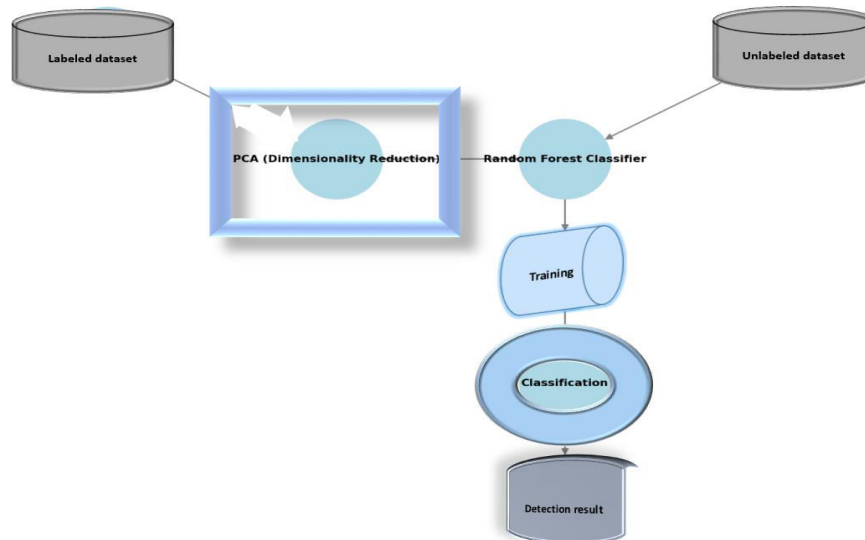
SYSTEM ARCHITECTURE

Fig 1: System Architecture

SYSTEM METHODOLOGY***Random Forest***

One well-known machine learning method that is a part of the supervised learning approach is Random Forest. In machine learning, it can be applied to both classification and regression issues. Its foundation is the idea of ensemble learning, which is the process of merging several classifiers to solve a challenging issue and enhance the model's performance. "Random Forest is a classifier that contains a number of decision trees on various subsets of the provided dataset and improves the predicted accuracy of that dataset by taking the average," as the name implies. Rather than depending on a single decision tree, the random forest forecasts the final result by taking the predictions from each tree and calculating the majority vote of predictions. Because there are more trees in the forest, accuracy is higher and overfitting is avoided. Leo Breiman and Adele Cutler created the popular machine learning algorithm Random Forest, which aggregates the output of several decision trees to produce a single outcome. Because it can handle both classification and regression problems, its versatility and ease of use have encouraged its use. This post will explain the random forest algorithm's operation, how it varies from other algorithms, and how to use it.

Algorithm

Step:1 The first step is to gather the data that you want to use to train the Random Forest

Step: 2 Next, you need to define the problem that you want to solve using Random Forest. In this case, it is a binary classification problem

Step: 3 To evaluate the performance of the Random Forest model, you need to split the data into training and test sets.

Step: 4 After training the Random Forest model, you can evaluate its performance on the test set.

Step: 5 If the performance of the Random Forest model is not satisfactory, you may need to tune the hyper parameters to achieve better results.

Step: 6 Once you are satisfied with the performance of the Random Forest model, you can deploy it to make predictions on new, unseen data.

SYSTEM MODULES

- Data Collection
- Dataset
- Data Preparation
- Model Selection
- Analyze and Prediction
- Accuracy on test set
- Saving the Trained Model

Module Descriptions

Data Collection:

This is the first real step towards the real development of a machine learning model, collecting data. This is a critical step that will cascade in how good the model will be, the more and better data that we get, the better our model will perform. There are several techniques to collect the data, like web scraping, manual interventions and etc.

Datasets:

The dataset consists of 125974 individual data. There are 42 columns in the dataset.

Feature name	Description	Type
Duration	Length(number of seconds)of the connection	continous
services	Network services Service on the destination,e.g., http,etc.	discrete
Src_bytes	Number of data bytesfrom source To destination.	continous

Dst_bytes	Number of data bytes from destination to source	continuous
Flag	Normal or error status of the connection	discrete
Land	1 if connection is from/to the same host/port;0 otherwise	discrete
Wrong_fragment	Number of “wrong” fragments	continuous
urgent	Number of urgent packets	continuous
Hot	Number of “hot” indicators	continuous
Num_failed_logins	Number of failed login attempts	continuous
Logged_in	1 if successfully logged in;0 otherwise	discrete
Num_compromised	Number of “compromised” conditions	continuous
Root_shell	1 if root shell is obtained;0 otherwise	discrete
Su_attempted	1 if “suroot” command attempt;0 otherwise	discrete
Num_file_creations	Number of file creation operations	continuous
Num_shells	Number of shell prompts	continuous
Num_access_files	Number of operations on access control files	continuous

Num_outbound_cmds	Number of outbound commands in an ftp session	continuous
Is_hot_login	1 if the login belongs to the "hot" list; 0 otherwise	discrete
Error_rate	% of connections that have "SYN" errors	continuous
Same_srv_rate	% of connections to the same service	continuous
Diff_srv_rate	% of connections to different services	continuous
Srv_count	Number of connections to the same services as the current connection in the past two seconds	continuous
Srv_error_rate	% of connections that have "SYN" errors	continuous
Srv_reject_rate	% of connections that have "REJ" errors	continuous
Srv_diff_host_rate	% of connections to different hosts	continuous
Num_root	Number of "root" accesses	continuous
Protocol_type	Type of the protocol, e.g. tcp, Udp, etc	discrete
Is_guest_login	1 if the login is a "guest" login; 0 otherwise	discrete

Data Preparation:

We will transform the data. By getting rid of missing data and removing some columns. First we will create a list of column names that we want to keep or retain. Next we drop or remove all columns except for the columns that we want to retain. Finally we drop or remove the rows that have missing values from the data set. Split into training and evaluation sets.

Model Selection:

The method that is employed, particularly for the reduction of the dimension of the provided dataset, is principal component analysis. One of the most effective and precise techniques for lowering the dimensionality of data is principal component analysis, which yields the intended outcomes. Using this technique, the features of the provided dataset are condensed into a predetermined number of characteristics known as principle components. With this approach, all of the input is treated as the dataset, which has a large number of attributes and a high dimension. By aligning the data points on the same axis, this technique shrinks the dataset. The primary components are calculated by shifting the data points along a single axis. The following procedures can be used to complete the PCA:

1. Take all dimensions d of the dataset.
2. Determine each dimension d 's mean vector.
3. Determine the dataset's overall covariance matrix.
4. Determine the eigen values ($v_1, v_2, \dots, v_3, \dots, v_d$) and eigen vectors ($e_1, e_2, e_3 \dots e_d$).
5. To obtain a matrix of $d \times n = M$, sort the eigenvalues in decreasing order and choose the n eigenvectors with the highest eigenvalues.
6. By creating a new sample area with this M form.
7. The primary components are the sample spaces that were obtained.

The first step involves creating the forest using the provided dataset, and the second step involves the classifier's prediction.

Saving the Trained Model:

Once you're confident enough to take your trained and tested model into the production-ready environment, the first step is to save it into a .h5 or .pkl file using a library like pickle. Make sure you have pickle installed in your environment. Next, let's import the module and dump the model into .pkl file

RESULTS AND DISCUSSION



Fig 2: Home Page

Fig 2 shows the Home Page where the User can click Login to use the application.

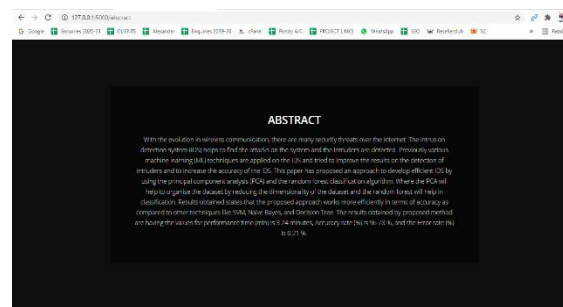


Fig 3: Abstract Page

Fig 3 shows the Abstract Page where the User can know about the application,

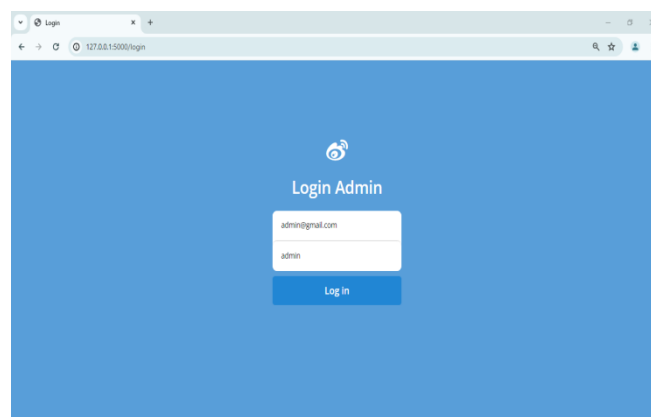


Fig 4: Login Page

Fig 4 shows the Login Page where the User can Login into the application using their id and password.

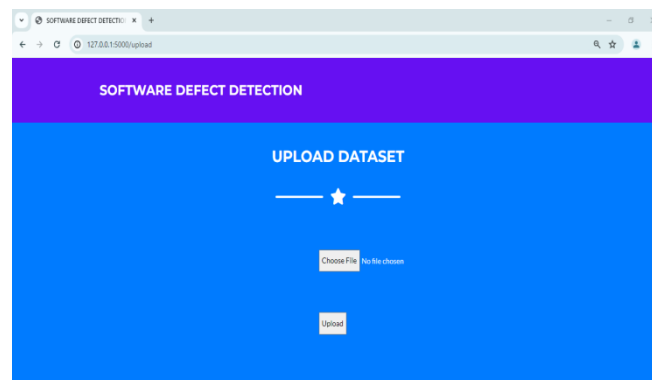
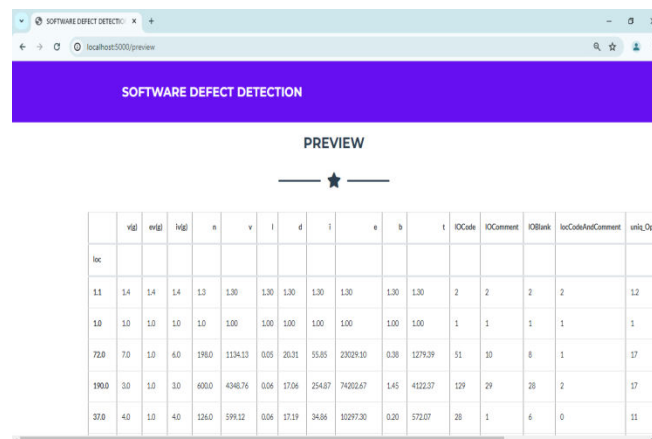


Fig 5: Upload Page

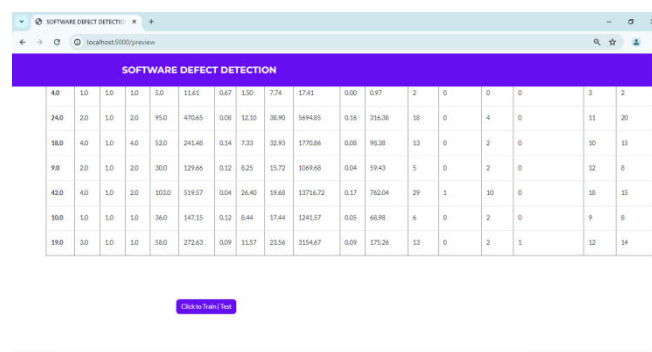
Fig 5 shows the Upload Page where the User can choose the dataset file and upload it.



	vid	enid	hgid	n	v	i	d	i	e	b	t	IOCode	IOComment	IOBank	locCodeAndComment	unlc_Op
loc																
11	14	14	14	13	130	130	130	130	130	130	2	2	2	2		12
10	10	10	10	100	100	100	100	100	100	100	1	1	1	1		1
720	70	10	60	1980	1134.53	0.05	20.31	55.85	23029.10	0.38	1279.39	51	10	8	1	17
1900	30	10	10	6050	4346.76	0.06	17.06	254.87	74202.67	1.45	4122.37	129	29	28	2	17
370	40	10	40	1260	599.12	0.06	17.19	34.86	10297.30	0.20	572.07	28	1	6	0	11

Fig 6: Preview Page

Fig 6 shows the Preview Page where the User can see the preview of the Dataset they uploaded.



	40	10	10	10	10	1165	0.67	1.50	7.74	17.41	0.00	0.97	2	0	0	0	3	2
240	20	10	20	95.0	470.65	0.06	12.10	38.90	5694.85	0.16	316.38	58	0	4	0		11	20
180	40	10	40	520	243.48	0.14	7.33	32.93	1770.86	0.08	98.38	53	0	2	0		10	15
90	20	10	20	360	129.66	0.12	6.25	15.72	1069.68	0.04	59.43	5	0	2	0		12	8
420	40	10	20	1030	519.27	0.04	26.40	59.68	13716.72	0.17	762.04	29	1	10	0		18	15
180	10	10	10	360	147.15	0.12	8.44	17.44	1241.57	0.05	68.98	6	0	2	0		9	8
180	30	10	10	580	272.63	0.09	11.57	23.56	3154.67	0.09	175.26	53	0	2	1		12	14

Fig 7: Train/Test Page

Fig 7 shows the Train/Test Page where the User click to train or test the uploaded dataset.

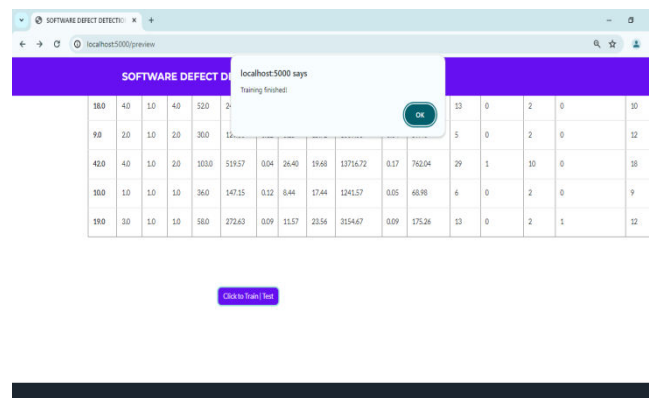
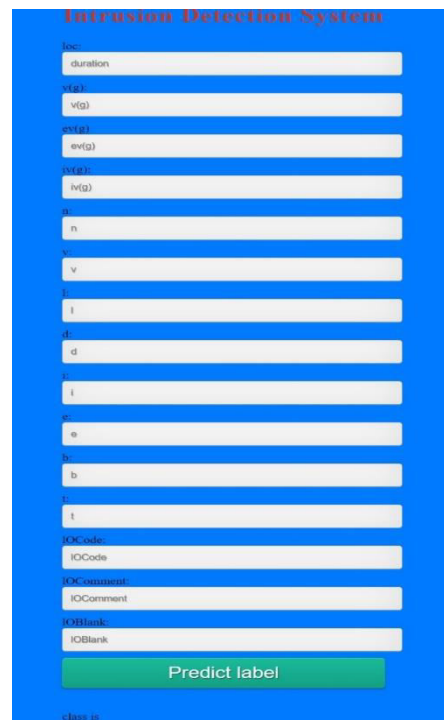


Fig 8: Training Complete Page

Fig 8 shows the Training Complete Page where the User gets notified that training is completed.



The screenshot shows a web application titled "Intrusion Detection System". It contains a form with the following input fields: duration, v(g), v1(g), ev1(g), v2(g), rv(g), n, m, v, f, d, f, e, b, i, IOC code, IOCCode, IOC comment, IOCComment, IOC link, and IOClink. A green "Predict label" button is at the bottom.

Fig 9: Predict Label Page

Fig 9 shows the Predict Label Page where the User can give the inputs and predict whether the software is ransomware clicking the Predict Label button

CONCLUSION

As the involvement of the systems over the internet increasing rapidly, the security concerns have also seen. The proposed approach deals with the detection of intruders over the internet efficiently. The proposed algorithm has performed well as compared to the previously applied algorithms such as SVM, Naïve Bayes, and Decision Tree. The detection rates and the false error rates can be improved at a great extent by the proposed approach. The dataset used here is the knowledge discovery dataset. The results

obtained by our proposed method having the values for Performance time (min) is 3.24 minutes, Accuracy rate (%) is 96.78 %, and the Error rate (%) is 0.21 %

REFERENCES

- [1] Jafar Abo Nada; Mohammad Rasmi Al-Mosa, 2018 International Arab Conference on Information Technology (ACIT), A Proposed Wireless Intrusion Detection Prevention and Attack System
- [2] Kinam Park; Youngrok Song; Yun-Gyung Cheong, 2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigData Service), Classification of Attack Types for Intrusion Detection Systems Using a Machine Learning Algorithm
- [3] S. Bernard, L. Heutte and S. Adam “On the Selection of Decision Trees in Random Forests” Proceedings of International Joint Conference on Neural Networks, Atlanta, Georgia, USA, June 14-19, 2009, 978-1-4244-3553-1/09 /2009 IEEE
- [4] Tesfahun, D. Lalitha Bhaskari, “Intrusion Detection using Random Forests Classifier with SMOTE and Feature Reduction” 2013 International Conference on Cloud & Ubiquitous Computing & Emerging Technologies, 978-0-4799-2235-2/13 2013 IEEE
- [5] Le, T.-T.-H., Kang, H., & Kim, H. (2019). The Impact of PCA-Scale Improving GRU Performance for Intrusion Detection. 2019 International Conference on Platform Technology and Service (PlatCon). Doi:10.1109/platcon.2019.8668960
- [6] Anish Halimaa A, Dr. K. Sundarakantham: Proceedings of the Third International Conference on Trends in Electronics and Informatics (ICOEI 2019) 978-1-5386-9439-8/19/2019 IEEE “MACHINE LEARNING BASED INTRUSION DETECTION SYSTEM.”
- [7] Mengmeng Ge, Xiping Fu, Naeem Syed, Zubair Baig, Gideon Teo, Antonio Robles-Kelly (2019). Deep Learning-Based Intrusion Detection for IoT Networks, 2019 IEEE 24th Pacific Rim International Symposium on Dependable Computing (PRDC), pp. 256-265, Japan.
- [8] R. Patgiri, U. Varshney, T. Akutota, and R. Kunde, “An Investigation on Intrusion Detection System Using Machine Learning” 978-1-5386-9276-9/18/ c2018 IEEE.
- [9] Rohit Kumar Singh Gautam, Er. Amit Doegar; 2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence) “An Ensemble Approach for Intrusion Detection System Using Machine Learning Algorithms.”
- [10] Kazi Abu Taher, Billal Mohammed Yasin Jisan, Md. Mahbubur Rahma, 2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST) “Network Intrusion Detection using Supervised Machine Learning Technique with Feature Selection.”
- [11] L. Haripriya, M.A. Jabbar, 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA) “Role of Machine Learning in Intrusion Detection System: Review”
- [12] Nimmy Krishnan, A. Salim, 2018 International CET Conference on Control, Communication, and Computing (IC4) “Machine Learning-Based Intrusion Detection for Virtualized Infrastructures”
- [13] Mohammed Ishaque, Ladislav Hudec, 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS) “Feature Extraction using Deep Learning for Intrusion Detection System.”
- [14] Aditya Phadke, Mohit Kulkarni, Pranav Bhawalkar, Rashmi Bhattad, 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC) “A Review of Machine Learning Methodologies for Network Intrusion Detection.”



- [15] Iftikhar Ahmad , Mohammad Basher, Muhammad Javed Iqbal, Aneel Rahim, IEEE Access (Volume: 6) Page(s): 33789 – 33795 “Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection.”
- [16] B. Riyaz, S. Ganapathy, 2018 International Conference on Recent Trends in Advanced Computing (ICRTAC)” An Intelligent Fuzzy Rule-based Feature Selection for Effective Intrusion Detection.”