International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-2, December, 2019

Network Malware Detection using Soft Computing and Machine Learning Techniques

Yogita R. Kulkarni, Sandeep A. Thorat

Abstract: ***In today's world there is rapid increase in the information which makes addressing of security issues more important. Malware detection is an important area for research in effective and secure functioning of computer networks. Research efforts are required to protect the systems from various security attacks. In this paper, we analyze usefulness of Soft Computing and Machine Learning Techniques for network malware detection. Hamamoto et al. [1] used combination of Genetic Algorithm and Fuzzy logic for implementation of network anomaly detection. The research work proposed in this paper extends the concepts discussed in [1]. The proposed work explores use of various Machine Learning algorithms such as K-Nearest Neighbor, Naïve Bayes and Decision Tree for network anomaly detection. The experimental observations are conducted on CIDDS (Coburg Intrusion Detection Data Set) dataset [14]. It is observed that Decision Tree approach gave better results as compared to KNN and Naïve Bayes techniques. Decision Tree technique gives 99% of accuracy and precision of 1 and recall of 1.***

Keywords: Network Malware Detection, Soft Computing, Machine Learning, K-Nearest Neighbors, Naïve Bayes, Decision Tree.

I. INTRODUCTION

Malware, or malignant software, is any program or document that is unsafe to a computer user [3]. Additionally, it is software which is intended to infiltrate a computer system without the proprietor's informed approval. For example, a program in the system which starts running without user's permission or a program which is monitoring user's personal information etc. Following are different threats due to malwares – Losing private and sensitive information, annoying advertisements, Encrypting the data and asking money to decrypt it etc. There are various types of malwares e.g. Virus, Worms, Trojan, Spyware, Cookies, Botnet, Sniffers, etc. [3].

Protection of confidentiality, integrity and availability in a computer network is a challenging task. [8]. A system which is able to find abnormal patterns or malicious traffic in network is known as Network Malware Detection system. For example, a malicious traffic in a network could mean that a hacked computer is conveying some sensitive information to an unauthorized host. To provide the security in a networking from unauthorized access network anomaly detection system is useful. A productive and successful mechanism is required to verify content. Multiple levels of data confidentiality in commercial and government organization need multi-layer protection. Parameter tuning is a large problem, especially in cases where malware only occur during testing. While you can apply the malware detection techniques for better performance, the biggest challenge is in making sure that the data itself is complete, accurate and consistent.

Network malware detection system will immediately separate the malicious events from the network data. There are two different methods are available to detect malware in the system [6]. These are signature-based detection and anomaly-based detection methods. There are also three different techniques for both detection methods. Static approach, dynamic approach and hybrid approach. These malware detection methods and techniques are shown in Fig.1 [15]. In Signature based method, malware is detected by signature information. Storage area such as repository is needed for storing signatures. Anomaly based method is used for detecting malware by monitoring of a system activity and classifying it as a malicious or normal. This system performs opposite of the signature-based system because it only detects attacks for which a signature has previously been created. Capturing and preprocessing of network traffic or data is an essential task to detect the network malware.

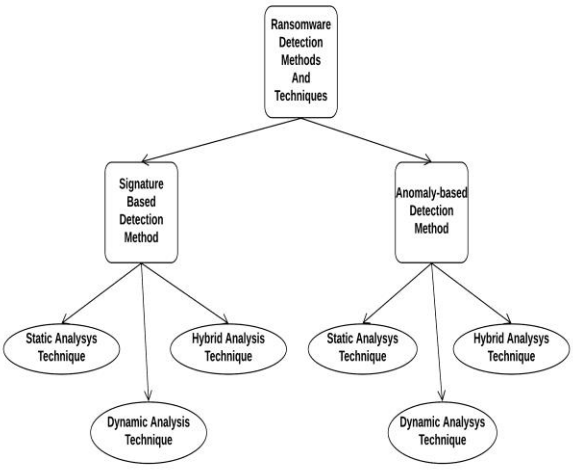

Fig.1. Malware Detection Methods and Techniques

Different datasets and tools have been used to capture as well as to analyze the network. The dataset includes benchmark dataset, KDDcup99 dataset, DARPA 2000 dataset, NSL KDD, synthetic dataset etc.

Revised Manuscript Received on December 30, 2019.
* Correspondence Author
Yogita R. Kulkarni, Department of Computer Engineering, Rajarambapu Institute of Technology, Rajaramnar, India Email: yogita.kulkarni@ritindia.edu
Dr.Sandeep A. Thorat, Department of Computer Engineering, Rajarambapu Institute of Technology, Rajaramnar, India Email: sandip.thorat@ritindia.edu

This paper includes analysis of some Soft Computing and Machine Learning algorithms which applied on a CIDDS-001 dataset. Soft computing deals with approximate models and gives solutions to difficult complex as well as real-life problems [1].It is mostly based on the techniques like genetic algorithm, fuzzy logic, artificial neural network etc. Additionally, there is allow computational cost for Soft Computing techniques addressing the network anomaly detection issue and also when compared to other methods [2,3]. Machine Learning is a major technique which plays an important role for network malware detection. It is a concept which allows the machine to learn from examples and experience. In machine learning there are mainly two tasks, training and testing. To detect the malicious events, machine learning requires complex and large datasets which includes different types of abnormal and normal patterns.

Here, in this paper some machine learning algorithms like K-Nearest Neighbor, Naïve Bayes and Decision tree have been implemented. The analysis of this work has been performed using performance metrics including Accuracy, Precision, Recall and F-measure etc. Decision tree gives best result in this experiment as compared to other algorithms.99% accuracy is given by decision tree. Key research contributions of this work are as follows:

- The research work provides an analysis of network malware detection done using machine learning and soft computing algorithms.
- The research work does fine tuning of the parameters used in machine learning algorithms; this helps to find out better results than previous work.
- This work gives performance criteria used for evaluating algorithms and systems for network malware detection.

The structure of paper is given as follows: Section II gives the literature review of this work. Section III discusses theoretical concept of Network Anomaly detection and machine learning methods. Section IV gives the experimental setup and result analysis using evaluation metrics. Section V explains Discussion. The paper ends by giving conclusions in the Section VI.

II. LITERATURE REVIEW

There are various techniques and approaches available for network malware detection system in literature which is shown in below section.

In 1949, computer pioneer John von Neumann explained the conceptual description of malware [22].The first malware was a virus called Creeper. It was created by engineer Bob Thomas. Therefore, by utilizing the different techniques malwares have been identified in a system. Anomalies are the unexpected patterns or events in a network. Chandola *et al.*[4] explains the types of anomalies and the different techniques to detect anomalies in network. Mostly, in networking area researchers have been developed different techniques, methods as well as algorithms to intelligently detect the network anomalies. Bhuyan *et al.*[5]discussed the criteria for the evaluation and classification for network intrusion detection systems. Taxonomy and some brief description of various existing datasets have been explained. The broad survey of Anomaly detection systems as well as Hybrid intrusion detection systems of recent past and present have given by Patcha *et al.*[6]while Zhang *et al.* proposed technique-based classification framework to categorize the normal or abnormal data. The techniques include in this paper was based on Statistical, Nearest Neighbor, Clustering, Classification and Spectral decomposition. Chauhan *et al.*[7] gives some most important types of anomaly detection technique and these are classification, clustering, information theory and statistical etc. Eskin *et al.*[8]provides framework for unsupervised anomaly detection technique.

In [5] and [15], authors give the techniques for malware detection system. Deepak Venugopal *et al.* [17] describe efficient signature-based malware detection on mobile devices. This system is effective in detecting identified malware. But this system cannot detect new malware which will come into system. In [18], for cyber security author discusses different data mining techniques. Author proposed a framework for traditional signature-based detection in malware detection system. In anomaly detection and data stream mining this research contains malicious code detection by mining binary executables. In [10]author gives the Signature based Intrusion Detection System(IDS) which is able to maintain integrity of data in network. But this system depends on the predefined intrusion patterns which are already created in system. In IDS if the signature database is not updated then the attacks present in network just passes through the system without being noticed. Some anomaly-based techniques are also present in literature. Hamamoto *et al.*[1] have proposed Network Anomaly Detection System by using Genetic Algorithm and Fuzzy logic. The work in this paper suggested an unsupervised training approach and the testing is done using flow-based data given by a campus network. While a novel fuzzy anomaly detection system designed using hybridization of PSO (Particle Swarm Optimization) and K-means clustering algorithm have been proposed by Karami *et al.*[11].In [23] author gives payload content based anomaly detection system. This system can identify the anomalous packet payload in network. Zhang *et al.*[12] proposed a novel algorithm ColSVM (Collaborative Support Vector Machine) which is based on semi supervised learning approach and independent component analysis.

There are different approaches for analysis of malware in a system[21]. In [20]author proposed a framework which has been developed for static as well as dynamic analysis of malicious events in a network. Machine learning model have been used in this system. J48decision tree gives the best performance using the terms of accuracy and precision. Santos et al. described a hybrid malware detector which uses a set of features generated from static and dynamic analysis of malicious code. For classification author consider various learning algorithms and these are K-Nearest Neighbor, Support Vector Machine, Decision Tree and Bayesian Network. This system describes that hybrid approach gives better performance than static and dynamic approaches.

The proposed work uses Soft Computing techniques including Genetic Algorithm and Fuzzy logic. Also, there is a use of Machine Learning approach including K-Nearest Neighbor, Naive Bayes and Decision Tree for malware detection system.

This work will analyze the performance of these techniques for detecting malicious activities in network flow.

III. PROPOSED NETWORK ANOMALY DETECTION USING MACHINE LEARNING TECHNIQUES

In the existing system [1] Genetic algorithm and Fuzzy Logic has used to classification of Malware. Basically, Genetic Algorithm is Optimization array algorithm which consist five different stages. Initially it generates a random population with the help of each genes and combination of multiple genes it generates single chromosome. Crossover has work for generate a new population and mutation has changed the random genes values from available genes. Fitness function is another stage to calculate the weight of each chromosome and according to descending order it apply selection operator on entire population. Finally, fuzzy algorithm generates the probability for each chromosome and generate the final rules, and these rules has used during classification of instances. So, here soft computing algorithms have been used for the classification of malicious or normal data in network. But the proposed system is extension of this existing system.

In proposed malware detection system machine learning is used for classification of normal or abnormal data from the collected data in network. In previous works, different machine learning algorithms have been applied in malware detection system [5]. This paper explains some machine learning algorithms, K-nearest neighbor, Naïve Bayes and Decision Tree.

A. K-Nearest Neighbor Classifier

It is a very simple, easy to understand, flexible and one of the topmost machine learning algorithms [15]. This classifier is based on distance function that basically measures the similarity or difference between two instances. Predictions can be made for the new instance(x) by looking through complete training set of the K most similar instances. It is most commonly utilized distance-based classifier and uses each training occurrences as a prototypical instance [16].The distance measure is used to determine which is the K-instance includes in training dataset most similar to new input. So, the most popular distance measure is Euclidean distance. It is calculated as follows,

$$D(x, x_i) = \sqrt{\sum_{i=1}^{n} \left(x_j - x_{ij}\right)^2} \quad (1)$$

Where, x is new point, x_i is existing points and j is input attribute. Therefore, the values for K can be establish by algorithm tuning. So, after applying this algorithm at last it will predict the classes like attacker, victim and normal for data.

B. Naïve Bayes Classifier

In literature, for binary and multiclass classification problems Naïve Bayes can be apply. The probabilities have represented in this algorithm. This includes, in training dataset the probability of each class is known as class probability and the conditional probability of each input value gives each class value. Here we apply Gaussian Naïve Bayes which is extension of Naïve Bayes. Other functions can be used to estimate the distribution of data but this Gaussian (normal distribution) is the easiest to work with because you only need to estimate mean and standard deviation from the training data. The mean and standard deviation values of each input variable(x) for each class value can be calculated as follows,

$$\mu = \sum_{i=1}^{n} x_i \quad (2)$$

Where, n is number of instances and x are values for input variables.

$$\sigma = \left[\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2\right]^{0.5} \quad (3)$$

Where, n is number of instances and $x_i$are values for input variables.

Using the Gaussian Probability Density Function Probability will be calculated as,

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma^2 y}\right) \quad (4)$$

Where $P(x_i|y)$ is the Gaussian Probability Density Function, $\sigma$is the standard deviation, $\pi$is the numerical constant ,exp is the numerical constant e or Euler's number raised to power and x is the input value for the input variable. So, Gaussian Probability Density Function will provide the probability of that new input value for that class

C. Decision Tree

In Machine Learning, Decision Trees[17] can be described also as the combination of mathematical and computational techniques to aid the description, categorization and generalization of a given set of data[18]. Data comes in records of the form:

$$(x, y) = (x_1, x_2, x_3, \ldots x_k) \quad (5)$$

The dependent variable y is the target variable that we are trying to understand, classify and generalize. The vector x is composed of the features, x_1, x_2, x_3 etc. Entropy based node splitting criteria had used for partitioning results. Entropy measures the homogeneity of a node and is defined as,

$$entropy = \sum_{j}^{n} p\left(\frac{j}{t}\right) log_2\left(\frac{j}{t}\right) \quad (6)$$

Where$p\left(\frac{j}{t}\right)$ is the relative frequency of class j at node t.

As we move downwards the tree, the level of uncertainty decreases, thus getting to a better classification or best split at every node some parameters have been used. To decide that, splitting measures Information Gain, gini index etc. is used. gini index is developed by the Italian statistician Corrado Gini. In decision trees gini index for a given node t is given as,

$$gini(t) = 1 - \sum_{j}^{n} \left[p\left(\frac{j}{t}\right) \right]^2 \quad (7)$$

Where $p\left(\frac{j}{t}\right)$ is the relative frequency of class *j* at node *t*.

In this experiment Dynamic anomaly detection method has been used. Fig.2 shows this detection method. In this method, it is essential to have the knowledge to decide whether the system is harmful. Rule sets are used to determine whether the system has established benign and valid behavior. But the anomaly-based detection systems will work steadily as long as the rule sets are well defined before. This method consists of two phases. These are "training/learning" and "monitoring/detecting" phases. During the training phase, the benign behaviors of the system are revealed and the patterns of the system are observed. After the benign behaviors are defined in the training phase, this method switches to "monitoring" phase and compares the real-time system with benign behaviors learned during the previous phase. The monitoring phase generates an alert or warning message when something excepting benign malicious behaviors occurs in the system.

IV. EXPERIMENTAL SETUP AND RESULTS

The below section discusses experimental environment, such as dataset and some hardware, software requirements for the experiment. It also gives information about performance measure format and the discussion about the results.

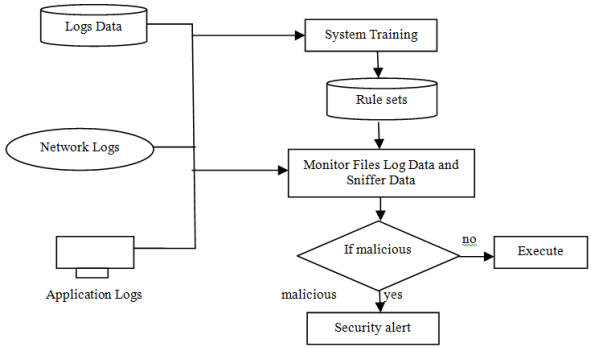

Fig. 2. Dynamic Anomaly Detection Method

A. Experimental Setup and Dataset

For the system performance evaluation, calculate the matrices for accuracy. The system is executed on INTEL 3.60 GHz i3 processor and 4 GB RAM with python 3.7 environment. As the dataset is large in size, it faces trouble to train the learning models by utilizing the CPU. Therefore, the models are trained on Google Collaborator. The CIDDS-001 dataset has used for proposed experiment analysis of system. CIDDS (Coburg Intrusion Detection Data Sets) consists of network flow based data which consist of 14 attributes [16]. Here in this experiment 6 attributes have been used. The dataset randomly splitting into training and testing. 67% data used for training purpose and 33% data used for testing purpose. Before, training the preprocessing process has been applied on dataset.

B. Experimental Setup and Dataset

Finally, performance of the system is evaluated based on various performance metrics such as accuracy, precision, recall and F-measure.[19] Accuracy refers to the closeness of measured value to a standard or known value. Precision and recall these are the success measure of prediction when the classes are very imbalanced. F-measure is a harmonic average of precision and recall. These metrics can be calculated as follows.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (8)$$

$$Precision = \frac{TP}{(TP + FP)} \quad (9)$$

$$Recall = \frac{TP}{(TP + FN)} \quad (10)$$

$$F - measure = 2 * \frac{(Precision * Reall)}{(Precision + Recall)} \quad (11)$$

Where,

TP is True Positive

TN is True Negative

FP is False Positive

FN is False Negative

C. Analysis Using Soft Computing Algorithms

The Table 1 shows the performance evaluation of system with Genetic Algorithm and fuzzy logic using CIDDS data set. It provides accuracy around 0.96 and F-score 0.84% respectively. Basically, Genetic Algorithm is the Optimization algorithm which is used in this experiment and the fuzzy logic used for classification to get the better probabilistic accuracy for entire system. The proposed system classifies each instance as Normal, Attack and Victim respectively.

D. Analysis Using Machine Learning Algorithms

i. Performance of K-Nearest Neighbors Classifier

The K-Nearest Neighbors method is implemented in five different ways, the different NN values has taken as input to execute the algorithm on entire data set. When number of NN has increased then system automatically increased

Table- I: Performance of Genetic Algorithm and Fuzzy Logic on CIDDS dataset

Algorithms	Precision	Recall	F-measure	Accuracy
Genetic Algorithm and Fuzzy	0.95	0.76	0.84	0.9653

the time complexity as well. For 4NN value, the accuracy obtained is 100%. Remaining all provides accuracy around 99.95% respectively. According to this experiment we conclude there is no big difference for accuracy with K-Nearest Neighbors size.

ii. Performance of Naïve Bayes Classifier

Naive Bayes has used as a second algorithm to evaluate the proposed system. The accuracy of Naive Bayes with CIDDS dataset is 81%, where higher F-measure for victim class prediction as well as lower for attack class its around 0.64. The highest precision provides for victim class as a 1 and lower for attack class it's around 0. 47. It provides Recall near about 0.98 as well as lower for victim class as 0.78.

iii. Performance of Decision Tree

The final experiment has done with Decision Tree (DECISION TREE) classification algorithm. The four different parameters have tune to get the performance of algorithm.

Table- II: Performance of Naïve Bayes on CIDDS dataset

Algorithm	Evaluation Metrics				
	Precision	Recall	F-measure	Class	Accuracy
Naïve Bayes	0.49	0.98	0.65	Normal	0.8160
	1.00	0.78	0.88	Victim	
	0.47	0.99	0.64	Attack	

criteria, *max_depth*, *max_feature*, *min_sample split* are the parameter which is used during the DECISION TREE execution. It's around provides 99% accuracy when *min_sample split* has value 2 as well as precision, recall and F-measure is having value1. While it provides accuracy for all classes 0.8696 when criteria have *gini*, *max_depth*=2, *max_feature*=1, *min_sample split*=10 and with *criteria* has entropy, *max_depth*=2, *max_feature*=1, *min_sample split*=10 accuracy should be 0.8696.

Table- III: Performance of K-Nearest Neighbors on CIDDS dataset

KNN	Evaluation Metrics			Class	Accuracy
	Precision	Recall	F-measure		
1NN	1.00	1.00	1.00	Normal	0.9995
	1.00	1.00	1.00	Victim	
	1.00	1.00	1.00	Attack	
2NN	1.00	1.00	1.00	Normal	0.9995
	1.00	1.00	1.00	Victim	
	1.00	1.00	0.99	Attack	
3NN	1.00	1.00	1.00	Normal	0.9995
	1.00	1.00	1.00	Victim	
	1.00	1.00	1.00	Attack	
4NN	1.00	1.00	1.00	Normal	1.0000
	1.00	1.00	1.00	Victim	
	1.00	1.00	0.99	Attack	
5NN	1.00	1.00	1.00	Normal	0.9994
	1.00	1.00	1.00	Victim	
	1.00	1.00	1.00	Attack	

The precision recall and F-measure will be highest for victim while noting for attacker.

The proposed system providing the Malware Detection System with 3 different machine learning classification algorithms. Each classifier having own execution strategy which provides higher results. The whole experiment analysis also illustrates accuracy, precision, recall as well as F-measure respectively. All three classification algorithms provide higher accuracy as compared to classical Genetic Algorithm and Fuzzy. The K-Nearest Neighbors provides accuracy around 99.95% with 1 for precision recall and F-Measure respectively. The Naive Bayes having around 81% accuracy. Decision tree provides around 99.98% accuracy for *gini* as well as *entropy* method with different values of *max_depth*, *max_feature*, *min_sample split.*

V. DISCUSSION

Many classification algorithms in machine learning have proposed in existing approaches. The proposed system is completely supervised learning according to illustrated system architecture. The purpose behind to chosen given three different classification algorithms is that each having a better accuracy than other machine learning classifiers.

Basically K-Nearest Neighbors should we take more time to generate the different clusters according to given K values, but it probably provides higher accuracy then other classification algorithms. Similarly, Naive Bayes classifier creates runtime probability according to the train model. Basically, it's a complete vector-based classifier which shows the results based on available background knowledge. It also provides good accuracy then other classification algorithms.

The third classifier we have carried out as a decision tree, it is the part of random forest algorithm when system generate the final result according to current probabilistic weight. This classifier is also better than traditional machine learning algorithms as well as regression-based classifiers.

VI. CONCLUSION

The proposed system illustrates execution of Malware Detection as well as classification using different Machine Learning Algorithms. Three machine learning algorithms have been proposed to detect the malicious activity from network data set based on the extracted features.

In each experiment illustrated, performance evaluation is calculated using statistical parameters. The Decision Tree provides 99% accuracy with Precision, Recall and F-measure having values 1 respectively. The Naive Bayes gives 81% accuracy. K-Nearest Neighbors also provides near about 99% accuracy.

The future work will be to implement the proposed research on various network dataset like a real time as well as synthetic data with different machine learning as well as deep learning algorithms. Distributed environment execution, it will be the interesting part for future direction of the system.

Table- IV: Performance of Decision Tree on CIDDS dataset

	Parameters				Evaluation Metrics			Class	Accuracy
	criteria	max_depth	max_feature	min_sample split	Precision	Recall	F-measure		
1	gini	None	None	2	1.00	1.00	1.00	Normal	0.9998
					1.00	1.00	1.00	Victim	
					1.00	1.00	1.00	Attacker	
	entropy	None	None	2	1.00	1.00	1.00	Normal	0.9998
					1.00	1.00	1.00	Victim	
					1.00	1.00	1.00	Attacker	
2	gini	2	1	10	0.69	0.84	0.75	Normal	0.8696
					0.89	0.96	0.92	Victim	
					0.00	0.00	0.00	Attacker	
	entropy	2	1	10	0.68	1.00	0.81	Normal	0.8772
					0.91	0.95	0.93	Victim	
					0.00	0.00	0.00	Attacker	
3	gini	12	1	20	1.00	1.00	1.00	Normal	0.9997
					1.00	1.00	1.00	Victim	
					1.00	1.00	1.00	Attacker	
	entropy	12	1	20	1.00	1.00	1.00	Normal	0.9997
					1.00	1.00	1.00	Victim	
					1.00	1.00	1.00	Attacker	
4	gini	22	1	30	1.00	1.00	1.00	Normal	0.9998
					1.00	1.00	1.00	Victim	
					1.00	1.00	1.00	Attacker	
	entropy	22	1	30	1.00	1.00	1.00	Normal	0.9998
					1.00	1.00	1.00	Victim	
					1.00	1.00	1.00	Attacker	
5	gini	32	1	40	1.00	1.00	1.00	Normal	0.9998
					1.00	1.00	1.00	Victim	
					1.00	1.00	1.00	Attacker	
	entropy	32	1	40	1.00	1.00	1.00	Normal	0.9998
					1.00	1.00	1.00	Victim	
					1.00	1.00	1.00	Attacker	

REFERENCES

1. A. H. Hamamoto, L. F. Carvalho, L. D. H. Sampaio, T. Abrão, and M. L.Proença, "Network Anomaly Detection System using Genetic Algorithm and Fuzzy Logic," Expert Syst. Appl., vol. 92, pp. 390–402, 2018.
2. M. Geden and J. Happa, "Classification of malware families based on runtime behaviour," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics),vol. 11161 LNCS, pp. 33–48, 2018
2. W. Mao, Z. Cai, D. Towsley, Q. Feng, and X. Guan, "Security importance assessment for system objects and malware detection," Comput. Secur., vol. 68, pp. 47–68, 2015.
3. V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," ACM Comput. Surv., vol. 41, no. 3, pp. 1–72, 2009.
4. M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network Anomaly Detection: Methods, Systems and Tools - IEEE Journals & MaGenetic Algorithmzine," vol. 16, no. 1, pp. 303–336, 2014
5. A. Patcha and J. M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," Comput. Networks, vol. 51, no. 12, pp. 3448–3470, 2007.
6. Y. Zhang, N. Meratnia, and P. HavinGenetic Algorithm, "Outlier detection techniques for wireless sensor networks: A survey," IEEE Commun. Surv. Tutorials, vol. 12, no. 2, pp. 159–150, 2010.
7. P. Chauhan and M. Shukla, "A review on outlier detection techniques on data stream by using different approaches of K-Means algorithm," Conf. Proceeding - 2015 Int. Conf. Adv. Comput. Eng. Appl. ICACEA 2015, pp. 580–585, 2015.
8. A. Karami and M. Guerrero-Zapata, "A fuzzy anomaly detection system based on hybrid PSO-Kmeans algorithm in content-centric networks," Neurocomputing, vol. 149, no. PC, pp. 1253–1269, 2015.
9. K. Zhang, C. Li, Y. Wang, X. Zhu, and H. Wang, "Collaborative Support Vector Machine for Malware Detection," Procedia Comput. Sci., vol. 108, pp. 1682–1691, 2015.
10. I. Firdausi, C. Lim, A. Erwin, and A. S. Nugroho, "Analysis of machine learning techniques used in behavior-based malware detection," Proc. - 2010 2nd Int. Conf. Adv. Comput. Control Telecommun. Technol. ACT 2010, pp. 201–203, 2010.
11. H. M. Deylami, R. C. Muniyandi, I. T. Ardekani, and A. Sarrafzadeh, "Taxonomy of malware detection techniques: A systematic literature review," 2016 14th Annu. Conf. Privacy, Secur. Trust. PST 2016, pp. 629–636, 2016.
12. F. Nelli and F. Nelli, "Machine Learning with scikit-learn," in Python Data Analytics, 2018.
13. A. Verma and V. RanGenetic Algorithm, "Statistical analysis of CIDDS-001 dataset for Network Intrusion Detection Systems using Distance-based Machine Learning," Procedia Comput. Sci., vol. 125, pp. 709–716, 2018.
14. S.Safavian,D.Systems,U.Man et al. "A Survey of Decision tree Classifier Methodology," vol. 21, no. 3, 1991.
15. R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, and S. Venkatraman, "Robust Intelligent Malware Detection Using Deep Learning," IEEE Access, vol. 7, pp. 46715–46738, 2019.
16. Deepak Venugopal, G.H., "Efficient signature-based malware detection on mobile devices" Mob. Inf. Syst., 2008. 4(1): p. 33-49.
17. D. Kumar, Patel and S. Bhatt., Implementing Data Mining for Detection of Malware from Code.Compusoft, 2014.3(4): p. 732-737
18. De Ocampo, Frances Bernadette C. and Del Castillo, Trisha Mari L., "Automated signature creator for a signature-based intrusion detection system with network attack detection capabilities" 2013.
19. Sethi, Kamalakanta, et al. "A Novel Malware Analysis Framework for Malware Detection and Classification using Machine Learning Approach." Proceedings of the 19th International Conference on Distributed Computing and Networking. ACM, 2018.
20. Santos, I., Devesa, J., Brezo, F., Nieves, J. and Brin Genetic Algorithms, P.G. (2013) OPEM: A Static-Dynamic Approach for Machine Learning Based Malware Detection. Proceedings of International Conference CISIS'12-ICEUTE'12, Special Sessions
21. Tian, Ronghua "An integrated malware detection and classification system." No. Ph. D. Deakin University, 2011.
22. S.A. Thorat, A.K. Khandelwal, B. Bruhadeshwar, K. Kishor," Payload content-based network anomaly detection," 1st International Conference on the Applications of Digital Information and Web Technologies, ICADIWT 2008, pp.127-132,2008.

AUTHORS PROFILE

Yogita R. Kulkarni is a M.Tech Scholar at department of Computer Science and Engineering, Rajarambapu Institute of Technology with specialization in Machine Learning. Area of interest Networking, Data Mining etc.

Dr.Sandeep A. Thorat did Ph.D. from Shivaji University Kolhapur, India. He completed M.Tech. from IIIT Hyderabad with specialization in Information Security. His area of interest is Machine Learning, Wireless Networks and Information Security. He has authored a book on C programming and designed a MOOCs course on Udemy platform.