International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/CSEIT217538

Survey and Performance Analysis of Machine Learning Based Security Threats Detection Approaches in Cloud Computing

Rajesh Keshavrao Sadavarte[1], **Dr. G. D. Kurundkar** [2]
[1]Assistant Professor and Head, Netaji Subhashchandra Bose College, Nanded, Maharashtra, India
[2]Assistant Professor, Computer Science Department, Shri. GuruBuddhiswami Mahavidyalaya, Purna District Parbhani, Maharashtra, India

ABSTRACT

Article Info
Volume 7, Issue 6
Page Number: 49-58

Publication Issue :
November-December-2021

Article History
Accepted : 10 Nov 2021
Published : 22 Nov 2021

Cloud computing is gaining a lot of attention, however, security is a major obstacle to its widespread adoption. Users of cloud services are always afraid of data loss, security threats and availability problems. Recently, machine learning-based methods of threat detection are gaining popularity in the literature with the advent of machine learning techniques. Therefore, the study and analysis of threat detection and prevention strategies are a necessity for cloud protection. With the help of the detection of threats, we can determine and inform the normal and inappropriate activities of users. Therefore, there is a need to develop an effective threat detection system using machine learning techniques in the cloud computing environment. In this paper, we present the survey and comparative analysis of the effectiveness of machine learning-based methods for detecting the threat in a cloud computing environment. The performance assessment of these methods is performed using tests performed on the UNSW-NB15 dataset. In this work, we analyse machine learning models that include Support Vector Machine (SVM), Decision Tree (DT), Naive Bayes (NB), Random Forests (RF) and the K-Nearest neighbour (KNN). Additionally, we have used the most important performance indicators, namely, accuracy, precision, recall and F1 score to test the effectiveness of several methods.

Keywords : Cloud Computing, Machine Learning, Cloud Security

I. INTRODUCTION

Recently the use of cloud computing has become increasingly popular. The personalised data centres have become popular as an inexpensive infrastructure solution for business plans. Cloud computing offers a wide variety of resources in the form of Internet services. Cloud computing assists users/organizations in reducing infrastructure costs by providing various online resources. Infrastructure as a service (IaaS), Platform as a service (PaaS) and Software as a service (SaaS) are still widely distributed and used by end-users. In this way, users do not need the knowledge, control and ownership of the cloud computing infrastructure and do not need to manage or control the infrastructure to deploy their applications.

Instead, they simply access or rent hardware or software that pays for only what they use. The possibility to pay as you proceed with the activities extensively demanded by cloud hosting providers is gaining popularity in the business-computing model [1].

Although Cloud computing is seen as a significant infrastructure change, more security work is still needed to reduce its failures. Since a significant amount of personal and corporate information is stored in cloud data centres, those cloud security and vulnerability issues need to be identified and prevented. Because cloud infrastructure uses standard Internet protocols and virtualisation techniques, it may be vulnerable to attack. Such attacks can come from traditional sources such as Address Resolution Protocol, IP spoofing, Denial of Service (DoS) [2], [3]. The traditional techniques used for detection and prevention do not work well enough to manage those attacks while also working with large data flows. Machine learning (ML) techniques are helpful in detecting attacks. Several solutions based on machine learning have been suggested to detect cloud attacks. A major challenge in machine learning-based solutions is to detect these attacks with high accuracy. The main purpose of this paper is to provide a comparative study and performance analysis using various techniques based on the study of machine learning techniques in cloud computing. We are analysing machine learning strategies by Random Forest (RF), K-Nearest Neighbours (KNN), Decision Tree (DT), Naïve Bayes (NB), and Support Vector Machine (SVM). For analytical purposes, UNSW-NB15 [4], [5] is used as a dataset and Python is used as a programming language.

The rest of the paper is structured as follows: Section II presents a literature review of the latest techniques used to detect the threat. Section III discusses machine learning methods, Section IV discusses data set, implementation and test results. Finally, the conclusion of the study is provided in Section V.

II. LITERATURE SURVEY

This section describes the Machine learning approaches-based threat detection systems.

Moustafa et al. [10] suggested a Collaborative Anomaly Detection Framework (CADF) handle big data in cloud computing. Provide technical services and how to deploy this framework in these areas. The proposed method consists of three modules: capturing and logging network data, pre-processing of this data, and a new decision engine using the Gaussian Mixture Model [20] and the lower–upper Interquartile distance limit [16] to detect attacks. The UNSW-NB15 database was used to test the new Decision Engine to test its reliability while modelling in real cloud computing systems and was compared to three ADS strategies. The design of using this mode as Software as a Service (SaaS) is designed for easy installation in cloud computing.

Osanaiye et al. [19] proposed An ensemble-based multi-filter feature selection method. This method achieves a good selection by combining the output of four filter methods. The proposed method has been used to use cloud computing and is used to detect DDOS attacks. Extensive experimental testing of the proposed method was performed using a database of intrusion detection benchmark, NSL-KDD, and decision-tree classifier. The results obtained indicate that the proposed method reduces the number of features to 13 instead of 41 well. Besides, it has a higher level of classification accuracy compared to other classification techniques.

Mobilio et al. [9] introduced Cloud-based anomaly detection as a service that uses a standard rule used in cloud systems to declare control of the concept of incorrect discovery. They also propose first results with lightweight machines that show a promising solution to better control the concept of detection of malformations. They also discussed how to apply the as-service paradigm to the unfavourable acquisition concept and gain anonymous acquisition as a service. They also recommend building a paradigm that supports paradigm as a service and can work in conjunction with any viewing system that stores data in a series of time series. Preliminary testing of as-a-service with the Clearwater cloud system obtained results showing how the as-a-service paradigm can effectively manage detection logic. This approach is interesting, incorporating new technologies for the use of unconventional real-time detection.

Aldribi et al. [21] introduced a hypervisor-based cloud for IDS that includes how to extract a novel feature based on user status functions and their hypervisor-related behaviour. The proposed model was intended to detect misconduct in the cloud following mathematical sequences using a collection of gradient descent and E-Div algorithms. The new database is presented as an intrusion detection database collected in a cloud that is also publicly available to investigators. The database includes multistage attack scenarios that allow for the development and testing of cloud computing threats. They performed experimental tests using the Riemann rolling feature extraction system and produced promising results. The database carries several connections over encrypted channels, for example, using protocols such as SSH.

Zhang [27] introduced multi-view learning strategies for detecting cloud computing platform inefficiencies using an explicit ML model. They work with a gap created as two phases in real-time, which is trained by developing many features of the ELM model. The presented technology automatically integrates many features from different sub-systems and finds an improved separation solution by reducing training errors. Conflict calculated between Sum is indicated by the relationship between the samples and the separation boundary, and the weighted samples set the recurrence rate of the separation model. The proposed model faces a variety of challenges in detecting inaccuracies, such as distribution imbalances, high-magnitude features, etc., well with Multi-view learning and feed control.

Fernandez and Xu [24] presented a case study using the Deep learning network to find out the threat. The author said he had achieved excellent results in detecting network threats. They also showed that using only the first three octets of IP addresses can be effective in managing the use of dynamic IP addresses, representing the DNN uncommon occurrence of DHCP. This approach has shown that autoencoders can be used to detect inaccuracies wherever they are trained in the expected flow.

Kwon [20] proposed Recurrent Neural Network RNN and Deep Neural Network DNN with ML mechanisms related to malformed network detection. They also performed local tests that demonstrated the feasibility of a DNN method for network traffic analysis. This survey also investigated the effectiveness of DNN models in network traffic analysis by introducing research into their FCN model. This approach demonstrates encouraging results with the accuracy of the development of threat findings compared to standard ML strategies, such as SVM, random forest, and Ad growth.

Garg et al. [23] introduced a hybrid data processing model for network malfunction detection that affects the performance of Gray Wolf Optimisation and Convolution Neural Network. The development of GWO and CNN training methods has been enhanced by testing initial capabilities and retrieval performance failures. These other expanded methods are called Improved-GWO and Improved-CNN. The proposed model operates in two phases of network threat detection. In the first phase, the enhanced GWO used feature selection to find the best trade between the two objectives to reduce the failure rate and reduce the feature set. In the second phase, advanced CNN is used for the separation of network threats. The author said the effectiveness of the proposed model was tested with a benchmark (DARPA'98 and KDD'99) datasets. They demonstrated the results obtained, confirming that the proposed cloud-based threat detection model was better than other related functions used for network anomaly detection. The proposed model shows a complete improvement of 8.25%, 4.08%, 3.62% in detection rate, false positives, and accuracy, respectively, related to the standard GWO and CNN.

Nisioti et al. [22] presented a study on the unsupervised model of the IDS. Features of this model are extracted from various sources of evidence such as network traffic, logs from different devices. Unsupervised techniques are proposed to be considered as flexible in the additional features extracted from various sources of evidence and do not require repeated training. They also suggested and compared the options for selecting IDS features. This survey finds and uses the lower set of features for each class to reduce computer time and stress.

Peng et al. [29] introduced IDS based on the decision tree algorithm. The authors compared the result of the work in many ways it was not only 10% of the database; all databases checked. Test results showed that the proposed IDS system was effective. However, compared to the detection time for each method, the decision tree time was not the best in the case of guaranteed accuracy. The authors argue that the proposed IDS system could be used in fog-computing environments in addition to big data. The proposed program was not tested as a real-time program. The program has used the older version of KDD cup 99, a newer, more recent version with significant improvements.

Manna and Alkasassbeh [31] have introduced the latest ML method, such as the J48 decision tree, the random forest, and the REP tree. The proposed process used SNMP-MIB data for the IDS-trained system to detect DOS attacks that could affect the network. Classifiers and features are used in the IP group. The results showed that using the REP tree algorithm provided the highest performance at IP set times. The performance between these three algorithms was accurate enough to be an IDS system. However, it is limited to the fact that the database has expanded and requires more real-time challenges.

Rathore and Park [8], have proposed a method based on a combination of extreme learning machines and a semi-supervised fuzzy c-means algorithm. ELM is trained using a training database and the membership rate of samples of unlabelled data is calculated using semi-supervised c-means. Samples with a higher membership value than the confidence level were further subdivided using ELM. In ELM classification, samples divided with higher confidence than the ELM confidence scale were included in the training database. This process continues until all unlabeled data samples are labelled.

Myint and Meesad have proposed a method known as the incremental learning algorithm based on SVM [11]. In this case, predictions are made using SVM and will reduce the steps required for calculation and complexity of the algorithm, error set, and time is saved for repeated data training. In this way, the author has used the KDD Cup99 dataset to evaluate system performance. The proposed system can predict 41 aspects of incoming data.

Nabila Farnaaz and M. A. Jabbar raised the model using the Random Forest to detect intrusion [12]. In this way, the author views the RF as the ensemble classifier and the model offers better performance compared to another traditional classifier of attack classification. To test the performance of the model, the author used the NSL-KDD dataset, and the proposed model works well with a low level of false alarm and a high level of detection.

Majjed et al. promote an effective and comprehensive STL-IDS deep learning approach that supports a self-taught learning framework [13]. With feature learning and size reduction, a suggested system can be used. In this way, To get high predictive accuracy of SVM training and testing time is reduced. The proposed method provides an improvement in network threat detection.

Sandhya Peddabachigari et al. examine the decision tree for intrusion detection [14]. This model was tested with the 1998 DARPA database, and the system offers better performance compared to traditional models with accuracy. Also, the results show that the training time and testing time are better compared to the support vector machine.

Mrutyunjaya Panda and Manas Ranjan Patra proposed a framework for NIDS based on the Naïve Bayes [15]. The implementation of KDD Cup 99 is used as a database and from the results, it is determined that the planned system offers high performance in terms of false-positive rate, process time and price.

Machine learning approaches [6]

Machine learning includes a series of algorithms that can learn patterns from data and predict accordingly. ML combines computer science and statistics to enhance prediction. ML comprises three main types of learning, supervised, unsupervised and semi-supervised. Supervised machine learning depends on classified data that are trained to build the classification model. Unsupervised learning algorithms enable training a model without guidance.

Naïve Bayes algorithm

The Naïve Bayes algorithm is used to perform classification, which is based on the Bayes theorem. This algorithm works on assumption that all input attributes are conditionally independent.

The steps of Naïve Bayes algorithm are as follows:

- Step 1: Given a training set S, Calculate the probability of each class $p(v_j)$.
- Step 2: Given a training set S, For each attribute value, a_i of each attribute a, calculate conditional probability $p(a_i|v_j)$.
- Step 3: Given an unknown instance X’, Classify X’ according to the best probability.

Decision Tree algorithm

Decision tree learning is a method for approximating discrete-valued target functions, in which the learned function is represented by a decision tree.

Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance. Each node in the tree specifies a test of some attribute of the instance, and each branch descending from that node corresponds to one of the possible values for this attribute. An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute in the given example. This process is then repeated for the subtree rooted at the new node.

The working steps of the Decision Tree algorithm are given below:

- Step 1: First, To place the best attribute from the dataset at the root of the tree some mathematical measure like information gain is used.
- Step 2: Second, Divide the training dataset into subsets. While dividing, we should consider each subset should contain data with the same value for an attribute.
- Step 3: Lastly, just repeat Step 1 and Step 2 on each subset until we find leaf nodes in all the branches of the tree.

Random forests algorithm

Random forests are an ensemble learning method for classification or regression that operate by constructing multiple decision trees by picking the “K” number of data points from the dataset and then merges them to get a more accurate and stable prediction. For each “K” data point’s decision tree, we have many predictions and then we take the average of all the predictions.

The steps for the Random Forest algorithm are as follows:

- Step 1: Select randomly “i” features from the entire “j” features with one condition i << j.
- Step 2: Using the concept of best split point, calculate node “n” from the “i” features.
- Step 3: Again using the concept of the best split, we need to split node “n” into daughter node.

- Step 4: Repeat Step 1–Step 3 until “1” number of nodes has been reached.
- Step 5: Build forest by repeating Step 1–Step 4 for “k” number of times to create “k” number of trees.
- Step 6: To predict the target, take test features and use the rules of each randomly created decision tree and store the predicted target.
- Step 7: Then simply find out votes for each predicted target.
- Step 8: At last, consider the high voted prediction target as a final prediction.

K-Nearest Neighbour algorithm

K-nearest neighbours (KNN) algorithm classifies new objects based on similarity measures. To measure similarity between different objects mathematical measure of Euclidean Distance is used. In the KNN algorithm, for each test data point, we would be looking at the K-nearest training data points and take the most frequently occurring classes and assign that class to the test data. Therefore, K represents the number of training data points lying in proximity to the test data point which we are going to use to find the class.

The steps of the K-Nearest Neighbours algorithm are given below:

- Step 1: Decide the value of K.
- Step 2: Calculate the distance between the query instance and all the training samples.
- Step 3: Sort the distance in ascending order and confirm nearest neighbours supported the Kth minimum distance.
- Step 4: Based on the majority of the class of nearest neighbours, assign the prediction value of the query instance.

Support Vector Machine (SVM)

The SVM classifier is used for classification and regression. In SVM, data is spat into the data point by using a hyperplane and it is used to determine the class of data point. The distance from the boundary to the nearest data point is called a margin and the data point that lies closest to the classification boundary is called a support vector. When we deal with SVM, then we have to assume two things: 1) The margin should be as large as possible, and 2) The support vectors are the most useful data points because they are the ones most likely to be incorrectly classified.

The working steps for SVM are as follows:

- Step 1: Define optimal hyperplane: maximize margin.
- Step 2: Extend the definition mentioned in Step 1 for nonlinearly separable problems: have a penalty term for misclassifications.
- Step 3: Map data to high-dimensional space where it is easier to classify with linear decision surfaces: reformulate problem so that data is mapped implicitly to this space.

Experimentation

We use UNSW-NB15 dataset to evaluate the effectiveness of threat detection methods designed using machine learning techniques. The tests were performed in Google Colaboratory under Python 3 using TensorFlow and Graphics Processing Unit (GPU).

Description of the database

The UNSW-NB-15 dataset was created using the IXIA PerfectStorm tool at UNSW Canberra's Cyber Range Lab to produce a hybrid of real modern normal activities and synthetic contemporary attack behaviour. The tcpdump tool was used to capture 100 GB of raw network traffic. These data contain nine types of attacks, namely, Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms. Argus, Bro-IDS network monitoring tools were used and twelve algorithms were developed to produce 49 features with a category label.

The total number of records is 2 million and 5,40,044 are stored in four CSV files, namely, UNSW-NB15_1. csv, UNSW-NB15_2. csv, UNSW-NB15_3. csv and UNSW-NB15_4. csv. The ground truth table is called UNSW-NB15_GT.csv and the event file list is called

UNSW-NB15_LIST_EVENTS.csv. The partition from this dataset was set up as a training set and a test set, i.e., UNSW_NB15_training-set.csv and UNSW_NB15_testing-set.csv, respectively. The number of records in the training set is 1,75,341 records and the test set consists of 82,332 records from various types, attacks and normal [4], [5].

Security threat detection methodology used in experimentation

The details of the threat detection methodology used in experimentation are illustrated in Fig. 1. Specifically, the method consists of four stages: (1) datasets stage, (2) pre-processing stage, (3) training stage and (4) testing stage.

Performance Metrics

We use the most important performance indicators, including, accuracy (ACC), recall (R), precision (P) and F1 score (F1). We can calculate the performance metrics using the following

Accuracy (ACC): It is a metric that is used to indicate the proportion of correct classifications of the total records in the testing set.

Accuracy = (TP+ TN)/ (TP+ FN+ TN+ FP)

Precision (P): It is a metric that measures the actual performance within the required answer space, i.e., among the positions.

P =TP/(TP + FP)

Recall (R): It is the metric by which we measure how much of the predicted answers are discarded or for every correct label, how many other true labels have we discarded.

R =TP/(TP + FN)

F1 Score (F): It is the harmonic mean of the two matrices P and R.

F =(2 ∗ P ∗ R)/(P + R)

Where,

True positive (TP): It can be outlined as anomaly instances properly categorized as an anomaly.

False-positive (FP): It can be outlined as normal situations wrongly categorized as an anomaly.

True negative (TN): It can be outlined as normal situations properly categorized as normal.

False-negative (FN): It can be outlined as anomaly instances wrongly categorized as normal. [6]

Results and Discussion

For comparison, five machine learning algorithms, namely, Support Vector Machine, Naive Bayes, Random Forests, Decision Tree, and K nearest neighbour were used. For comparison, evaluation parameters like accuracy, precision, recall and F1 score were considered and their comparison results are shown in Table I. We can say that the accuracy of the Naive Bayes algorithm is low and the accuracy of the Support Vector Machine algorithm is high.

1. COMPARISON OF MACHINE LEARNING BASED THREAT DETECTION MODELS

Algorithm	Accuracy (overall)	Precision		Recall		F1 Score	
		Attack	Normal	Attack	Normal	Attack	Normal
SVM	89.87	0.87	0.97	0.98	0.77	0.92	0.86
RF	89.49	0.86	0.97	0.99	0.76	0.92	0.86
KNN	88.23	0.84	0.96	0.98	0.74	0.91	0.84
DT	85.24	0.81	0.95	0.97	0.70	0.88	0.80
NB	47.89	0.23	1.00	1.00	0.38	0.38	0.55

III. Conclusion

Cloud computing offers a wide variety of resources in the form of Internet services. The smooth functionality of cloud services is essential to this technology. Attackers can use it to disrupt the performance of cloud services. In this work, comparative study and performance analysis of threat detection models is proposed for cloud computing using machine learning methods. The performance of the various models was assessed using the UNSW-NB15 dataset. The accuracy of the Naive Bayes algorithm is low and the accuracy of the Support Vector Machine algorithm is high. Through the literature survey, we understand the need to develop a comprehensive threat detection system using in-depth package testing on cloud computing.

IV. REFERENCES

[1]. S. Paul, R. Jain, M. Samaka, J. Pan, "Application Delivery in Multi-Cloud Environments using Software Defined Networking", Computer Networks Special Issue on cloud networking and communications, February 2014, pp. 166-186.

[2]. B. Xu, S. Chen, H. Zhang, and T. Wu, "Incremental k-NN SVM method in intrusion detection," in Proc. 8th IEEE Int. Conf. Softw. Eng. Service Sci. (ICSESS), Nov. 2017, pp. 712–717, doi: 10.1109/ICSESS.2017.8343013.

[3]. R. Moreno-Vozmediano, R. S. Montero, E. Huedo, and I. M. Llorente, "Efficient resource provisioning for elastic cloud services based on machine learning techniques," J. Cloud Comput., vol. 8, no. 1, p. 5, Dec. 2019, doi: 10.1186/s13677-019-0128-9.

[4]. M. Nour, J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," Military Communications and Information Systems Conference (MilCIS), IEEE, 2015.

[5]. M. Nour, J. Slay, "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," Information Security Journal: A Global Perspective, 2016, pp.1-14.

[6]. P. R. Chandre, P. N. Mahalle, and G. R. Shinde, "Deep Learning and Machine Learning Techniques for Intrusion Detection and Prevention in Wireless Sensor Networks: Comparative Study and Performance Analysis", Lecture Notes in Networks and Systems 82, https://doi.org/10.1007/978-981-13-9574-1_5

[7]. BADER ALOUFFI, MUHAMMAD HASNAIN, HASHEM ALYAMI, MUHAMMAD AYAZ, ABDULLAH ALHARBI, WAEL ALOSAIMI, "Systematic Literature Review on Cloud Computing Security: Threats and Mitigation Strategies", IEEE Access, 2021.

[8]. Rathore S , Park J H, "Semi-supervised learning based distributed attack detection framework for IoT", Appl. Soft Comput. 2018;72:79–89 .

[9]. Mobilio M, Orrù M, Riganelli O, Tundo A, Mariani L., "Anomaly detection as-a-service", In: 2019 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW). IEEE; 2019. p. 193–9.

[10]. Moustafa N, Creech G, Sitnikova E, Keshk M., "Collaborative anomaly detection framework for handling big data of cloud computing", In: 2017 military communications and information systems conference (MilCIS). IEEE; 2017. p. 1–6.

[11]. Myint, H. O., & Meesad, P., "Incremental Learning Algorithm based on Support Vector Machine with Mahalanobis distance (ISVMM) for Intrusion Prevention", 978-1-4244-33889/09/$25.00 ©2009 IEEE, (2009).

[12]. Farnaaz, N., & Jabbar, M. A., "Random forest modelling for network intrusion detection system", Procedia Computer Science, 89, 213–217 (Elsevier), (2016).

[13]. Al-Qatf, M., Lasheng, Y., Alhabib, M., & Al-Sabahi, K. (2018), "Deep learning approach combining sparse auto encoder with SVM for network intrusion detection", IEEE Access. https:// doi.org/10.1109/ACCESS.2018.2869577.

[14]. Peddabachigari, S., Abraham, A., & Thomas, J. (2016), "Intrusion detection systems using decision trees and support vector machines",

International Journal of Advanced Networking and Applications, 07(04), 2828–2834. ISSN: 0975-0290.

[15]. Panda, M., & Patra, M. R., "Network intrusion detection using Naïve Bayes", IJCSNS International Journal of Computer Science and Network Security, 7(12), (2007, December).

[16]. Peel D, McLachlan G J, "Robust mixture modelling using the t distribution", Stat Comput. 2000;10(4):339–48.

[17]. Van, N. T., Thinh, T. N., & Sach, L. T., "An anomaly-based network intrusion detection system using deep learning", In 2017 International Conference on System Science and Engineering (ICSSE).

[18]. Yang, Y., Zheng, K., Wu, C., Niu, X., Yang, Y., "Building an Effective Intrusion Detection System Using the Modified Density Peak Clustering Algorithm and Deep Belief Networks", Appl. Sci. 9, 238 (2019).

[19]. Osanaiye O, Cai H, Choo KKR, Dehghantanha A, Xu Z. Dlodlo M, "Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing", EURASIP J Wirel Commun Netw. 2016;1:130–130 (2016), https://doi.org/10.1186/s13638-016-0623-3.

[20]. Kwon D, Kim H, Kim J, Suh SC, Kim I, Kim K J, "A survey of deep learning-based network anomaly detection", Cluster Comput. 2019;22(1):949–61.

[21]. Aldribi A, Traoré I, Moa B, Nwamuo O., "Hypervisor-based cloud intrusion detection through online multivariate statistical change tracking", Comput Secur. 2020;88:101646–101646. https://doi.org/10.1016/j.cose.2019.101646.

[22]. Nisioti A, Mylonas A, Yoo PD, Katos V. ,"From intrusion detection to attacker attribution: a comprehensive survey of unsupervised methods", IEEE Commun Surv Tutor. 2018;20(4):3369–88. https://doi.org/10.1109/comst.2018.2854724.

[23]. Garg S, Kaur K, Kumar N, Kaddoum G, Zomaya AY, Ranjan R., "A hybrid deep learning-based model for anomaly detection in cloud datacenter networks", IEEE Trans Netw Serv Manage. 2019;16(3):924–35. https://doi.org/10.1109/tnsm.2019.2927886.

[24]. Fernández G C, Xu S, "A case study on using deep learning for network intrusion detection", In: MILCOM 2019–2019, IEEE Military Communications Conference (MILCOM). IEEE; 2019. p. 1-6.

[25]. Nicholas Lee, Shih Yin Ooi and Ying Han Pang, " A Sequential Approach to Network Intrusion Detection", Lecture Notes in Electrical Engineering 603, https://doi.org/10.1007/978-981-15-0058-9_2

[26]. Kishor Kumar Gulla, P. Viswanath, Suresh Babu Veluru, and R. Raja Kumar, " Machine Learning Based Intrusion Detection Techniques", Handbook of Computer Networks and Cyber Security, https://doi.org/10.1007/978-3-030-22277-2_35

[27]. Zhang J, "Anomaly detecting and ranking of the cloud computing platform by multi-view learning", Multimedia Tools Appl. 2019;78:30923–42.

[28]. Barbhuiya S, Papazachos Z, Kilpatrick P, Nikolopoulos DS, "RADS: Real-time anomaly detection system for cloud data centres", 2018, arXiv preprint arXiv:1811.04481.

[29]. Peng K, Leung VCM, Zheng L, Wang S, Huang C, Lin T, "Intrusion detection system based on decision tree over big data in fog environment", Wireless Commun Mob Comput. 2018;2018:1–10. https://doi.org/10.1155/2018/4680867.

[30]. Sapna S. Kaushik, Dr. Prof. P. R. Deshmukh, "Detection of Attacks in an Intrusion Detection System", International Journal of Computer Science and Information Technologies, Vol. 2 (3), 2011, 982-986.

[31]. Manna A, Alkasassbeh M., "Detecting network anomalies using machine learning and SNMP-

MIB dataset with IP group", In: 2019 2nd International Conference on new Trends in Computing Sciences (ICTCS). IEEE; 2019. p. 1–5.

[32]. Gopal Singh Kushwah, Virender Ranga, "Optimized extreme learning machine for detecting DDoS attacks in cloud computing", In computers & security 105 (2021) 102260. https://doi.org/10.1016/j.cose.2021.102260

[33]. Kashif Naseer Qureshi, Gwanggil Jeon, Francesco Piccialli, "Anomaly detection and trust authority in artificial intelligence and cloud computing", In Computer Networks 184 (2021) 107647. https://doi.org/10.1016/j.comnet.2020.107647

[34]. Fargana J. Abdullayeva, "Advanced Persistent Threat attack detection method in cloud computing based on autoencoder and softmax regression algorithm", In Array 10 (2021) 100067. https://doi.org/10.1016/j.array.2021.100067

[35]. S. Krishnaveni, S. Prabakaran, "Ensemble approach for network threat detection and classification on cloud computing", Concurrency Computat Pract Exper. 2019;e5272, https://doi.org/10.1002/cpe.5272

[36]. ALI BOU NASSIF, MANAR ABU TALIB, QASSIM NASIR, HALAH ALBADANI, FATIMA MOHAMAD DAKALBAB, "Machine Learning for Cloud Security: A Systematic Review", In IEEE Access, 2021.

[37]. Mohamed Amine Ferrag, Leandros Maglaras, Sotiris Moschoyiannis, Helge Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study", Journal of Information Security and Applications, 50 (2020) 102419.

Cite this article as :

Rajesh Keshavrao Sadavarte, Dr. G. D. Kurundkar, "Survey and Performance Analysis of Machine Learning Based Security Threats Detection Approaches in Cloud Computing", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 7 Issue 6, pp. 49-58, November-December 2021. Available at doi : https://doi.org/10.32628/CSEIT217538
Journal URL : https://ijsrcseit.com/CSEIT217538