COMPARATIVE ANALYSIS OF CLOUD IDS CLASSIFICATION ALGORITHMS

Arjun Choudhary Dept. of Comp. Sc. & Engg., MBM University, Jodhpur, India
a.choudhary@policeuniversity.ac.in
Dr. Rajesh Bhadada Professor and Head, Department of Electronics and Communication, MBM University, Jodhpur, India. rajesh_bhadada@rediffmail.com
Ram Prakash Prajapat SDE, BSNL, Jodhpur, India. prajapat@rediffmail.com

Abstract
Cloud computing is greater efficient and user friendly than traditional local computers., because it presents software, information, and big scale resources to every customer. One of the big problems is that conventional host-based cloud computing systems consume a lot of machine resources. In a cloud computing setting, HIDS (Host-Based Intrusion Detection System) has been proposed to safeguard virtual servers or instances. The basic purpose of the research presentation is to research wonderful approaches and models to seek out Denial of service attack (DoS) with the usage of host-based intrusions detection systems. There are precise prevention and detection models got by way of researchers makes use of unique supervised, unsupervised, and feature selection for detection and alarm generation of various threats in clouds virtual machines. In this paper, we ana-lysed different HIDS models on the basis of varied classifications like type, detection time, strategies, the dataset used, and detected DoS attack. NSL-KDD is used for the following purposes: Perform performance evaluation and analysis. Machine learning techniques such as Random Forest, KNN, and SVM were used in the experiments. The random forest is the best classification algorithm, with an accuracy of 99.81 percent and precision of 99.93 percent, respectively, and the proposed model is quite efficient and fast, according to the results of the testing.

Keywords: Intrusion, IDS, DoS, NSL-KDD, Random Forest, KNN, SVM

Introduction
Intrusion detection systems (IDS) are used to provide tools to detect system and net-work processes in the event of a security breach or malicious activity. IDSs are al-ways sorted by recognized media to form a NIDS (Network-based IDS) or a HIDS (host-based IDS) [11]. Where critical, IDS plays a protective role in asset classification and activation of protection mechanisms. The ultimate goal of an intrusion detection system is to prevent or reduce persistent attacks from cyber-attacks. Intrusion detection systems vary greatly in their detection approach and data source [5]. IDS can be divided into two models based on data source [16].

- Host-Based IDS (HIDS): It is a means of working over it by analyzing packets to specified hosts. HIDS is a model of IDS deployed on the host, either the DMZ or the user [2], for example, Web Server, in the form of logs and traffic of the host, can be analyzed.
- Network-Based IDS (NIDS): NIDS is always set at the entry (or gate) of the inter-net channel's network traffic [20]. NIDS analyses network TCP/IP data packets as part of its traffic analysis.

IDS is divided into two modules based on Detection Approaches [2]:

- Signature-Based Detection: By matching signatures or patterns with already exist-ed patterns in the database with the event (either intrusion or attack), the intrusion will be detected.
- Anomaly-Based Detection: for detecting an intrusion based on anomaly that occurs on the host or on the network, the premise profile that was created on IDS will be recognized as intrusion if it has abnormal behavior then the firstly created pro-file to define what normal activity in the host or network.[9].

Cloud computing offers consumers a variety of services via the Internet, including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). To address cloud security concerns, defending virtual computers and virtual networks from capacity attacks is a related option. IDS are placed in accordance with cloud architecture. The most significant and adaptable model for enabling IDS in cloud environments is with Infrastructure as a Service (IaaS). Denial of service

(DoS) attacks are one of the most serious security threats in the cloud. This limits the service's availability [26]. The correct IDS area in a comprehensive network system could be one of these types.

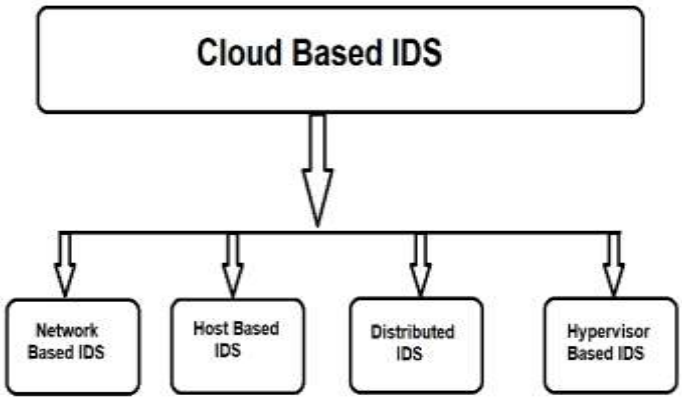

Figure 1. Various IDS Models

Cloud computing combines with a variety of well-known and advanced IT systems, technologies and networks. This is prone to security issues. As a result, cloud providers must completely protect their devices, both internally and externally, from intruders. At this, there are various protection technology [15] along with: authentication, signature, firewall, encryption, intrusion detection, access control. Most of these technologies are related to passive defense and intrusion detection. Intrusion detection initiates intrusions and generates alerts in a timely manner [8]. The second barrier once the firewall is intrusion detection, has become a hot space in recent year. Detecting harmful attacks and their varieties is one of the most significant and ambitious objectives of cloud computing. As cyber-attacks increase, it is essential to plan and implement a powerful intrusion detection system (IDS) to protect your information systems.

In this study, we use various classification methods such as Naive Bayes (NB), Decision Trees (DT), Random Forest (RF), SVM, and KNN to explore the dataset [25] of previous studies. The goal of this research is to determine which method has the best accuracy, precision, recall, and F-measure [6]. This research reveals that all of the aforesaid techniques perform admirably on system outputs across a variety of datasets.

This paper focuses on researching the best machine learning algorithms for Cloud IDS. The ML algorithms chosen in this study are Random Forest, KNN, and SVM. To train and validate a ML model, we use the NSL-KDD dataset. The features are extracted from the pre-processed dataset [18]. We then apply these capabilities to machine learning algorithms and compare their performance to denial-of-service attacks (DoS). The machine learning techniques utilized in intrusion detection systems are outlined in this article, as well as which networks produce the best anomaly detection. It also aids in the development of more precise and efficient detection systems for DoS attacks.

The rest of the paper is structured as follows: Related Activities The section II discusses, which provides other researchers with research related to machine learning and its applications. The Section III describes the dataset. Intrusion detection models proposed using various machine learning techniques are described in section IV. Implementation and experimental results are described in section V. This article ends with Section VI. This section contains conclusions and guidance for future work.

Related Work

In [24], R. Vijayanand proposed an IDS system by means of way of utilizing the CICIDS2017 data set and the Support Vector Machine Classifier (SVM). Seven extraordinary SVM models have been skilled for one-of-a-kind attacks (Port scan, DoS, Bot, Web Attack). For every model, he has used different 10-14 Genetic Algorithms, he got 99% accuracy rate with detection of attacks.

Imtiaz Ullah and Qusay H. Mahmoud [23] applied algorithm on NSL-KDD and ICSX datasets. In the research, it is based on feature-based selection and consequently the ranks are relegated to the features by using the classification process in that the information gain method is used. Within the research, operation performed with or without reducing the dimension and then comparative study has been done using those. As indicated thru result, accuracy of 41.55% is achieved using NSL-KDD dataset with Naive Bayes classifier, at the same time using Naive Bayes classifier, the quality detection system achieved an accuracy of 96.50. Comparing this method with the SVM classification Naive Bayes, J48 and the proposed method, we obtained the highest accuracy of 99.90% for the NSL-KDD dataset and 99.70% for the ISCX dataset.

Jianguo Jiang, Qian Yu et al. [10] performed detection of DDoS (ALDDoS) attacks using CICIDS2017 dataset in the Application Layer. They conducted two phases of research in terms of traffic levels and nodes. Traffic Levels: the request, Traffic Load, Average Rate, Average IP count, Average order Load, Number of Requests, Average Request Load, and IP Account Affinity Properties. The proposed model gives an accuracy of 99%.

In [1], Ibrahim al-Jamal et al. Hybrid IDS deployed in the cloud using UNSW-NB15 data set. Step 1/Clustering uses a k-means cross validation algorithm to classify network flows into 64 clusters. At this point, it is divided into (1) normal events and (2) abnormal events. The second step relies on a Support Vector Machine (SVM) algorithm to learn monitored recognition patterns. At this point, the trained SVM model recognizes new event anomalies. In the proposed approach, we observe the overall performance of each stage of the system and estimate the classification unit using the output data of the previous stage. The maximum accuracy of the
SVM model is 84.7%, which is suitable for a hybrid model. The maximum accuracy of the SVM model is 84.7%, which is sufficient for a hybrid model.

Vishal Sharma et al. [21] analysed and detect DDoS attacks using machine learning techniques in cloud computing environments. They used the tool Tor Hammer to conduct a DDoS attack on my private cloud. DDoS attacks detect access to IDS by sending suspicious generic objects to the server. The Snort generated database is classified into WEKA using Naive Bayes, SVM and Random Forest algorithms. Of these, SVM follows a random forest algorithm to provide the best results in terms of accuracy, precision, recall and measurement.

Unlike previous cloud-based intrusion detection solutions provided by efficient and independent algorithm selection. Therefore, this paper is a good place to compare the range of current solutions based on machine learning techniques. Using NSL-KDD Dataset and several machine learning algorithms the evaluation is done.

Dataset Description

Previously, after DARPA 98[13], the KDDcup99 [22] dataset was used to analyze intrusion behavior, however the dataset has some statistical flaws that end in poor assessment of anomaly detection. The internal transformation of the KDD data set results in a complex version of the NSL-KDD data set [19]. It is very difficult to reference an existing core network, but it can still be used as a reference record. An effective detection method [7] that identifies different intrusions for researchers to compare. The statistical studies show that data sets present fundamental problems that can significantly affect system performance and lead to very low estimates of how anomalies are detected. To address these issues, there are new datasets like NSL-KDD [7]. We recommend that you include specific data records for the entire KDD dataset.

Advantages

The following are some of the benefits of using the NSL-KDD dataset: [19]

- Since there are no duplicated records in the training set, the classifier does not produce skewed results.
- In the test set, there is no further evidence of improved percentage decrease.
- Measures the number of records in each surface composite record in relation to the original KDD dataset's record ratio.

The NSL-KDD training set consists of approximately 4,900,000 individual communication vectors, each containing 41 common or aggressive traits of a specific type.

Proposed Work

We first discuss the methods used in this proposed work, and then describe these methods and then apply them to build a detection model for high performance of intrusion detection system.

A. Overview of the framework

The proposed system uses Random Forest, K-Nearest Neighbor (KNN), and Support Vector Machine (SVM) algorithms. Figure 2 shows the framework implementation process.

The proposed framework detects Denial of service attack. The following stages comprise the development of the suggested model:

1. Importing the dataset: The proposed model was evaluated using the NSL-KDD dataset.
2. Pre-processing: This step is necessary to manage lost data. Batch processing of data is also part of pre-processing because the dataset is very large and some attributes are not suitable for use in machine learning algorithms.
3. Splitting the dataset into training and testing datasets: A 4:1 ratio was introduced with 80% of the original dataset used for training and the remaining 20% of the original dataset used for testing purpose.
4. Feature Scaling: It is used to transform various criteria into a single standard parameter to facilitate machine learning algorithms.

We can now insert our dataset into your machine learning algorithm. We used the scikit-learn library random forest classifier, k-nearest neighbor classifier, and Support Vector Classifier to classify feature sets and evaluate their performance.

B. Random Forest Classifier

Random Forest (RF) is a relatively new classification algorithm developed by Leo Breiman [17] that uses an independent set of classification or regression trees. Random forests generate many classified trees. Each tree is created as a separate boot-strap instance of the original data using a tree classification algorithm. Once the forest is created, we place and rank the newly sorted objects under each tree in the for-est. Each tree is chosen to reflect the feature class decision tree. Forest chooses the category that gets the most votes for the object. This is a random forest using bagging and boosting [12] as random variables to form a tree in a successful method. The main features of the random forest algorithm are:[17][6]

- It is unique among existing data mining algorithms in terms of accuracy.
- The element works efficiently on large feature-rich data sets and can provide key feature estimates.
- This factor has no nominal data problems, is highly disproportionate, unbalanced, and can handle unbalanced datasets.
- This factor provides an internally neutral estimate of the generalization error for the entire forest.

C. K-Nearest Neighbors Classifier

The k-Nearest Neighbors (kNN) method was used for statistical reasoning and pat-tern recognition as a non-parametric technique [12] in the early 1970s. Classification of k-neighbors is based on the k-

nearest neighbors of the sample being classified. The number 'k' is a user-defined integer that provides unsupervised and supervised learning capabilities.

The most important features of the K-Nearest Neighbors are:[12]

- Real-time prediction does not necessitate any prior knowledge. As a result, KNN algorithms are much faster than other training-based algorithms like SVM and linear regression.
- Since the algorithm need not to make predictions before training, it is easy to add new data.
- The implementation of the KNN requires only two parameters: a k value and a distance function (such as Euclid or Manhattan).

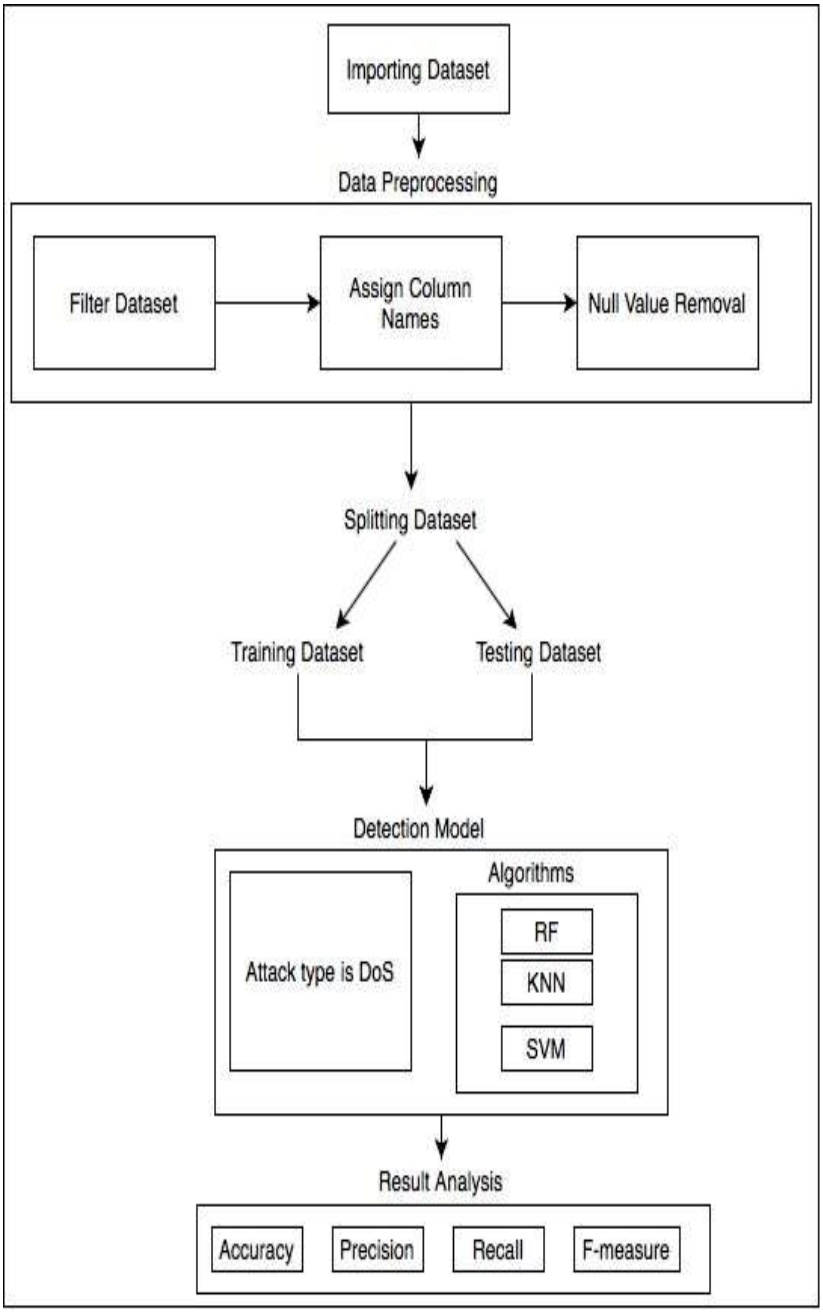

Figure 2. Proposed Classification Model

D. Support Vector Classifier

Support Vector Machines (SVM) are based on supervised machine learning algorithms that can be used to classify or regress statistical learning theory and were developed in 1962 by Russian scientist

Vladimir Naumovich Vapnik [4], which is used for classification problems. The SVM algorithm plots each data item as the closest point in n-dimensional space corresponding to n features. Here, each object has a specified coordinate value. Then we find and classifies the exceptional level by performing classification by finding the hyper-plane that clearly separates these two classes.

The main features of the support vector machines are:[12][4]

- The effect is effective in large spaces.
- It is also effective when the dimension number exceeds the sample number.
- It is also memory efficient because it uses a subset of decision function training points (called support vectors).
- Versatile: You can define some basic functions for decision making. Specific kernels are provided, but it is also possible to specify a custom kernel.

Experimental Results

The performance of the trained model depends on how the data is distributed. It may not represent normality of the model. For this, we used the k-fold cross-validation method. The data is first divided into equal-sized segments or folds. Since k folds are trained and tested with k iterations, we skip the experiment once per iteration for testing and train the model the rest of the k-1 folds [4]. We then average the accuracy from each iteration to get the model accuracy. The same is true for precision, recall and F-measure. This method has the advantage of allowing all observations to be used for both training and validation, with each observation only being used once for validation.

Performance: After running the selected classification algorithms, the summarised outcomes are there in the following table.

Table 1. Performance Results of Various Algorithms on NSL KDD Dataset

ML Classifier	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)
Random Forest	99.814	99.933	99.692	99.772
KNN	99.715	99.678	99.666	99.672
SVM	99.371	99.107	99.450	99.278

The random forest technique was employed to generate the best planning outcomes, as shown in Table I. The Random Forest Algorithm is the algorithm with the best accuracy, precision, recall, and f-measure. The proposed model correctly classifies all datasets and achieves 99.81%, 99.9%, 99.69% and 99.77% as accuracy, precision, recall, and F-measure according to the random forest approach. It is interesting to note that the accuracy of KNN is 0.34% better than that of SVM. Also, the random forest accuracy of 0.1% second is higher than the KNN accuracy.

After selecting a feature, the results do not change significantly. This suggests that not all suggested features contribute to performance improvement. Therefore, it is relatively more efficient and can improve the numerical character of the study. The proposed system provides higher accuracy. In other words, the model more accurately defines the proposed DoS attack (with or without feature detection).

ROC (Receiver Operating Characteristic) curve This graph shows the performance of the classification model.

For different classification thresholds, these curves represent TPR versus FPR. Increasing both True and False Positives [3], lowers the classification threshold, this classifies more items as positive.

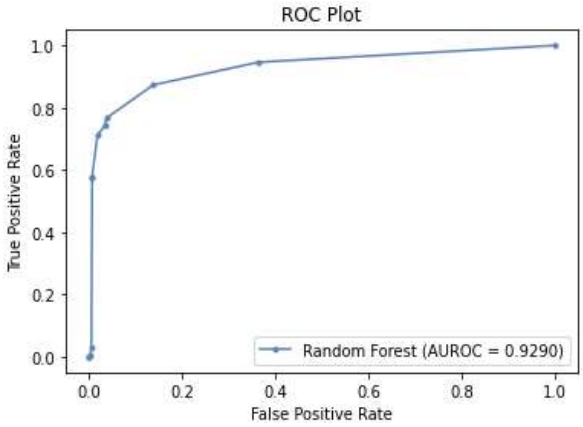

Figure 3. ROC curve for Random Forest model

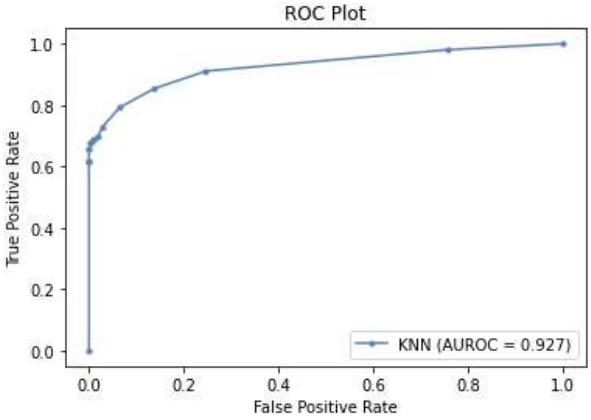

Figure. 4. ROC curve for KNN model

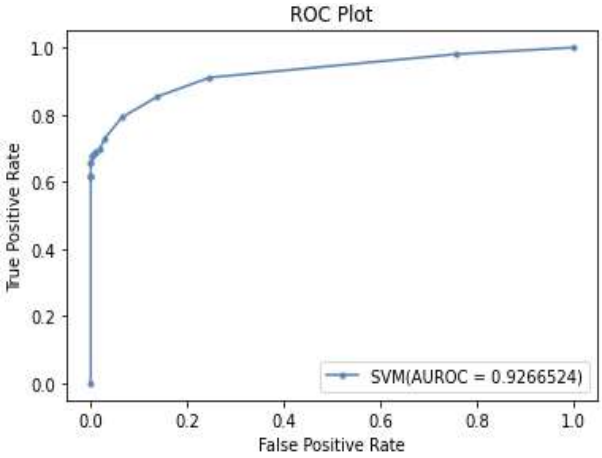

Figure 5. ROC curve for SVM model

These two parameters of the graph are: TPR (True Positive Rate) is synonym of recall, and FPR (False Positive Rate) are:

True Positive Rate = True Positives / (True Positives + False Negatives) (1)

False Positive Rate = False Positives / (False Positives + True Negatives) (2)

AUC/AUROC (Area Under the curve of the Receiver Operating Characteristic) Value

We can summarise the model skills as AUC value. We classify randomly selected positives and selected negatives by placing a randomly selected positive instances over a randomly selected negative instances [14], and this probability of classifier is also represented by AUC value. AUC ranges [0,1]. The higher the AUROC value, the better the classifier is.

The AUROC value of Random Forest classifier is highest among all, hence it is the better among KNN and SVM. The AUROC of KNN exceeds by AUROC of SVM by 0.004, and RF exceeds by 0.002. The AUROC of the perfect classifier is 1, and the classifier with AUROC greater than 0.5 is better than the random classifier. The obtained results show that the proposed model is sufficiently effective.

Conclusion

Using various ML approaches, we looked at some of the recent HIDS studies. As cloud attacks become more common and complex, new ways to improve cloud IDS must be found. In this paper, to detect the DOS attack using NSL-KDD dataset we give an overview of the review and proposed an architecture, using comparative performance of Random Forest, KNN and SVM algorithms for classification using accuracy, precision, recall, and F-measure. In the experiment we found that out of these, the Random Forest classifier has the highest performance as it has an accuracy of 99.81% for classification with less error rate to detect DoS attack. The simulations can combine different machine learning algorithms to make them more suitable for future tasks. One can use the Random Forest algorithm and algorithms that work with intrusions to detect cloud intrusions and further improve the accuracy of your classifier and can stop its Potential using Intrusion Prevention System (IPS).

References

1. Hybrid intrusion detection system using machine learning techniques in cloud computing environments. In 2019 IEEE 17th international conference on software engineering research, management and applications (SERA) (pp. 84-89). IEEE.
2. Besharati, E., Naderan, M., & Namjoo, E. (2019). LR-HIDS: logistic regression host-based intrusion detection system for cloud environments. Journal of Ambient Intelligence and Humanized Computing, 10(9), 3669-3692.
3. Brownlee, J., How to Use ROC Curves and Precision-Recall Curves for Classification in Python. Jan. 2021. URL: https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/.
4. Chandrasekhar, A. M., & Raghuveer, K. (2013, January). Intrusion detection technique by using k-means, fuzzy neural network and SVM classifiers. In 2013 International Conference on Computer Communication and Informatics (pp. 1-7). IEEE.
5. Chintala, S., & LeCun, Y. (2016). A path to unsupervised learning through adversarial networks. Facebook Engineering (June 20, 2016).
6. Deshpande, P., Sharma, S. C., Peddoju, S. K., & Junaid, S. (2018). HIDS: A host-based intrusion detection system for cloud computing environment. International Journal of System Assurance Engineering and Management, 9(3), 567-576.
7. Dhanabal, L., & Shantharajah, S. P. (2015). A study on NSL-KDD dataset for intrusion detection system based on classification algorithms. International journal of advanced research in computer and communication engineering, 4(6), 446-452.
8. Gautam, R. K. S., & Doegar, E. A. (2018, January). An ensemble approach for intrusion detection system using machine learning algorithms. In 2018 8th International conference on cloud computing, data science & engineering (confluence) (pp. 14-15). IEEE.
9. Jaber, A. N., Zolkipli, M. F., Shakir, H. A., & Jassim, M. R. (2017, November). Host based intrusion detection and prevention model against DDoS attack in cloud computing. In International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (pp. 241-252). Springer, Cham.
10. Jiang, J., Yu, Q., Yu, M., Li, G., Chen, J., Liu, K., ... & Huang, W. (2018, August). ALDD: a hybrid traffic-user behavior detection method for application layer DDoS. In 2018 17th IEEE International Conference on Trust, Security and Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE) (pp. 1565-1569). IEEE.
11. Khalid, F., Hanif, M. A., Rehman, S., & Shafique, M. (2018, December). Security for machine learning-based systems: Attacks and challenges during training and inference. In 2018 International Conference on Frontiers of Information Technology (FIT) (pp. 327-332). IEEE.
12. Masetic, Z., & Subasi, A. (2016). Congestive heart failure detection using random forest classifier. Computer methods and programs in biomedicine, 130, 54-64.
13. MIT lincoln MIT. DARPA dataset. Mar. 2021. URL: https://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusiondetection-evaluation-dataset.

14. Mma. How to plot ROC curve and compute AUC by hand. Oct. 2019. URL: https://mmuratarat.github.io/2019-10-01/how-to-compute-AUC-plot-ROC-by-hand.
15. Modi, C. N., & Acha, K. (2017). Virtualization layer security challenges and intrusion detection/prevention systems in cloud computing: a comprehensive review. the Journal of Supercomputing, 73(3), 1192-1234.
16. Shukla, D. K., Kumar, D., & Kushwaha, D. S. (2021). Task scheduling to reduce energy consumption and makespan of cloud computing using NSGA-II. Materials Today: Proceedings.
17. Satpathy, S., Prakash, M., Debbarma, S., Sengupta, A. S., & Bhattacaryya, B. K. (2019). Design a FPGA, fuzzy based, insolent method for prediction of multi-diseases in rural area. Journal of Intelligent & Fuzzy Systems, 37(5), 7039-7046.
18. Mohan, P., Sundaram, M., Satpathy, S., & Das, S. (2021). An efficient technique for cloud storage using secured de-duplication algorithm. Journal of Intelligent & Fuzzy Systems, (Preprint), 1-12.
19. Revathi, S., & Malathi, A. (2013). A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection. International Journal of Engineering Research & Technology (IJERT), 2(12), 1848-1853.
20. Sharma, D. K., Shukla, D. K., Dwivedi, V. K., Gupta, A. K., & Trivedi, M. C. (2021). An efficient Makespan reducing task scheduling algorithm in cloud computing environment. In ICT Analysis and Applications (pp. 309-315). Springer, Singapore.
21. Sharma, V., Verma, V., & Sharma, A. (2019, June). Detection of DDoS attacks using machine learning in cloud computing. In International Conference on Advanced Informatics for Computing Research (pp. 260-273). Springer, Singapore.
22. Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009, July). A detailed analysis of the KDD CUP 99 data set. In 2009 IEEE symposium on computational intelligence for security and defense applications (pp. 1-6). IEEE.
23. Shukla, D. K., Dwivedi, V. K., & Trivedi, M. C. (2021). Encryption algorithm in cloud computing. Materials Today: Proceedings, 37, 1869-1875.
24. Vijayanand, R., Devaraj, D., & Kannapiran, B. (2018). Intrusion detection system for wireless mesh network using multiple support vector machine classifiers with genetic-algorithm-based feature selection. Computers & Security, 77, 304-314.
25. Wang, X., Jiang, W., & Luo, Z. (2016, December). Combination of convolutional and recurrent neural network for sentiment analysis of short texts. In Proceedings of COLING 2016, the 26th international conference on computational linguistics: Technical papers (pp. 2428-2437).
26. Xie, M., & Hu, J. (2013, December). Evaluating host-based anomaly detection systems: A preliminary analysis of adfa-ld. In 2013 6th international congress on image and signal processing (CISP) (Vol. 3, pp. 1711-1716). IEEE.

