ISSN: 0974-5823 Vol. 7 No. 6 June, 2022

International Journal of Mechanical Engineering

A Secure dynamic Multilevel Intrusion Detection System using Machine Learning

Mr.B.Senthil Kumar[1], Dr.M.S.Josephine[2], Dr. V.Jeyabalaraja[3]

[1] Research Scholar, Bharathiar University, Coimbatore

[2]Professor, Dr.MGR Educational and Research Institute

[3] Professor, Velammal Engineering College

Abstract

In recent years, an advance in security systems has become more essential to apply many researchapplications. There is an increase in demand to develop a good security detection system. The complexity of new attacks are increasing every day, therefore an efficient machine learning model needsthe best type of appeal for intrusion detection.In this paper, we implemented machine learning algorithms to perform dynamic multilevel intrusion classification systemsto attain good accuracy and perfection in detection.Network intrusions classification is performed by using machine-learningtechniques such as K- Nearest Neighbor (NN), Support Vector Machines (SVM), Random Forest (RF) to detect the system attacks without any prior information. Moreover, the proposed system was evaluated by using NSL Canadian dataset.

Keywords: Machine Learning, Intrusion Detection System, Classification, Security

1 Introduction

Cyber security has become one of the essential research in Technology 4.0. The intrusion detection is represented in a machine learning (ML)-based IDS by a feature set that includes the relevant event behaviour to enable the classification of benign and malicious events, present and future Intrusion Detection System (IDS) deployments must be capable of performing at such high-speed network bandwidths [1]. As a result, an ML model should be able to correctly identify subsequent events as long as they exhibit the same behaviour as those seen during the training phase.Because the current intrusion model has outlived its usefulness, a new one must be constructed when network traffic behaviour changes [2]. The storing of fresh network data content, the labelling of data content events, the extraction and selection of intrusion characteristics, the ML algorithm parameter optimization, detection model training, and model testing are all part of creating an intrusion model [3].

The main approach to defend against advanced threat attacks, network intrusion detection is facing more and more challenges. The traditional intrusion detection system based on feature detection has been used for a long time. Being limited by the scale and refresh rate of the database of predefinedsignatures, a signature based intrusion detection system is not able to detect all types of attacks, especially new attack variants [4]. To solve this problem, researchers have paid much attention to introducing other techniques in intrusion detection, and one way is to use machine learning techniques.

The current research provides an adaptive machine learning model that may combine the benefits of each technique for various types of data detection and obtain optimal outcomes through ensemble learning [5]. Machine learning has the advantage of combining the predictions of multiple base estimators to increase generalizability and robustness over a single estimator. To train our model, we used the NSL Canadian data set and some common algorithms including Support Vector Machines, Random Forests, and K- means algorithm [6]. The machine learning methods, which improve the intrusion detection effect significantly. They outperform many earlier study findings and offer promising application potential [7].

2 Related Work

Several studies have suggested that by selecting relevant features for an intrusion detection system, it is possible to considerably improve the detection accuracy and performance of the detection engine [8]. ANNBayesian Net-GR technique that means ensemble of Artificial Neural Network (ANN) and Bayesian Net with Gain Ratio (GR) feature selection technique. proposed a mutual information-based algorithm that analytically selects the optimal feature for classification. This mutual information-based feature selection algorithm can handle linearly and nonlinearly dependent data features. an effective deep learning approach, self-taught learning (STL)-IDS, based on the STL framework [9]. The proposed approach is used for feature learning and dimensionality reduction. It reduces training and testing time considerably and effectively improves the prediction accuracy of support vector machines (SVM) concerning attacks. Because a network conversation typically consists of numerous packets, a single event (e.g., a network packet) usually does not allow for the correct behaviourcharacterisation required for feature extraction (message flow) [10]. When numerous packets are evaluated together, network-based assaults just depart from typical behaviour. A single packet sent during a flood-based DDoS assault, for example, could be a normal client creating a connection or an attack if examined separately [11].

The intrusion detection system (IDS) is a method that analyses user behaviour in the system after the user has logged in to identify intruders. User behaviour in the computer is monitored by a host-based IDS, which can identify suspicious behaviour as an intrusion or regular behaviour. This paper describes how an expert system uses a collection of rules as a pattern recognised engine to detect intrusions [12]. The author presented a PIDE (Pattern Based Intrusion Detection) model, which is based on the SBID (Statistical Based Intrusion Detection) model that was previously deployed. The results of the experiments show that combining the SBID and PBID approaches results in a comprehensive intrusion detection system [13].

Experts analyse all possible intrusions or malicious activities and then convert them into conditional rules, which are then compared against logs (monitoring data) by inference modules of IDSs to recognise any form of intrusion. Using statistical approaches such as time series and Markov chains, researchers were able to detect illegal users based on their behavior [14]. For detecting intrusion, Haystack developed an outline framework to identify malicious use, leaking, pretext assaults, denial of service, effort to infiltrate, and access control of ID. Forrest detects an attack by diverting the sequence from the expected profile, which is examined using call orders [15].

Rules are created in one of two ways: by an expert or by a system that generates rules automatically and apply them to acquired data frequently. When a rule is triggered, it either sends an alert to the system administrator or performs some automatic actions, such as barring the user or terminating the session, and this will continue until all rules have been triggered [16]. When the rule is triggered, an alert for terminating the session and blocking the user account is generated. The study is initially focused on statistical analysis, however, it is not suitable for huge datasets. As a result, the existing system has a lot of flaws. To address the current difficulties, a new system is required that improves the signature adjunct IDS results utilising a combined hybrid technique [17].

A framework named DFEL to identify internet intrusion in the IoT environment to avert irreparable cyberattack damage. The authors demonstrated that DFEL not only improves classifiers' accuracy in predicting cyber attacks but also significantly reduces detection time. Adaptive boosting was used by Arivudainambi et al. [18] with nave Bayes as the weak (base) classifier. The important finding of the study is that they were able to improve detection accuracy while correctly determining the attack using a less amount of features.The author presented the HeTL framework and technique, which can find the common latent subspace of two separate attacks and learn an optimal representation that is invariant to changes in attack behaviours.

PROPOSED APPROACH

A.) *Intrusion Detection extraction observable from the Internet*

Intruder behaviour obtained from network data content is used by network-based intrusion detection systems to detect intruders. Packets or network logs, such as NetFlow records, for example, can be used to create network data content [19]. In general, a large number of network packets arrive in an unorganised fashion (big

data settings). In other words, before a NIDS engine can handle network packets, they must be preprocessed. Fields of interest must be identified and processed during preprocessing before they can be passed to a feature extraction module. The feature extraction module's purpose is to extract features; in fact, a features vector is a set of network behaviours [20]. The networking event's goal is to use a feature vector to characterisebehaviour, which can subsequently be analysed and categorised using a machine learning algorithm.

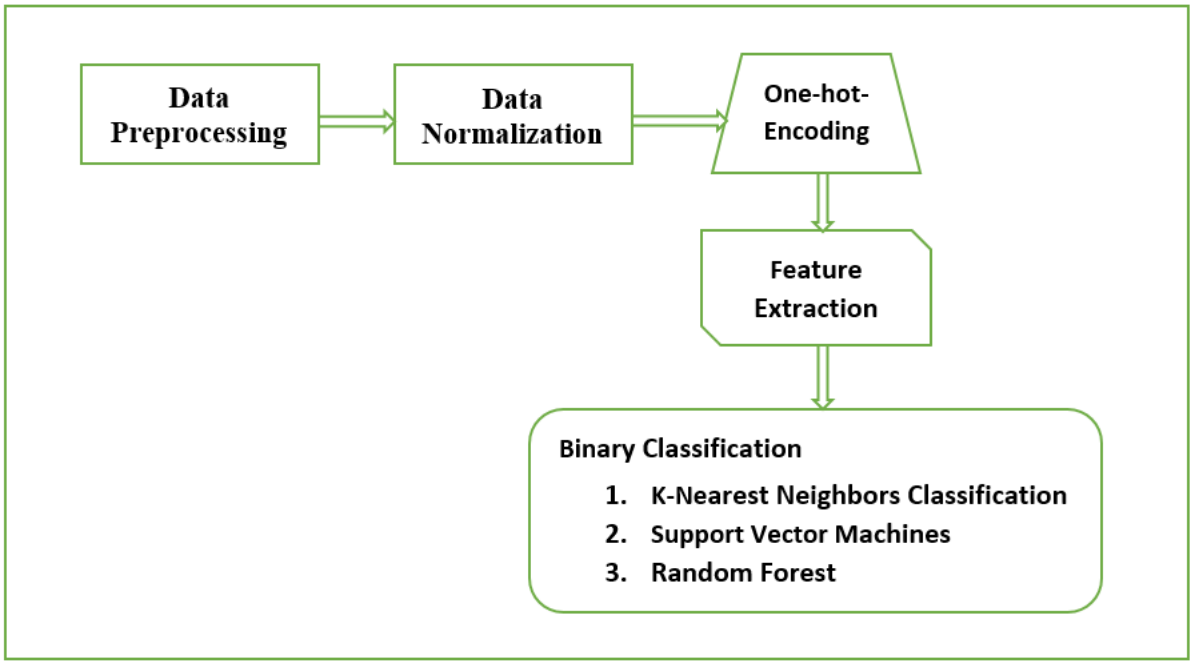

Figure 1 Proposed system Flow of Intrusion classification

B.) Network-Based Intrusion Detection Using Machine Learning

Generally, pattern recognition approaches are used to detect intrusions using ML-based algorithms, to classify a given input into a collection of classes [21]. Pattern recognition in NIDS is accomplished by categorising network material as either normal or attack. The effort of developing a classifier is divided into three phases: training, validation, and testing. The accuracy rates, such as the rates of true-positive, true-negative, false-positive, and false-negative occurrences, are measured when the model is evaluated using a test dataset [22]. The ratio of correctly classified examples to the total number of analysed instances is used to calculate the accuracy rate.

The ratio of attack events correctly classified is known as the true-positive (TP) rate, whereas the ratio of normal events correctly classified is known as the true-negative (TN) rate. A false-positive (FP) rate, on the other hand, refers to the proportion of normal events misclassified as attacks, whilst a false-negative (FN) rate refers to the proportion of attack events misclassified as benign. Several techniques have been proposed for the feature selection task, ranging from random subset selection to genetic search algorithms, which have yielded promising results [23]. The genetic search feature selection approach leverages the notion of gene selection to find the best subset of features

For instance, to overcome network-based intrusion detection challenges in high-speed environments, several works have proposed distributed and highly scalable intrusion detection mechanisms. In such a context, the data capturing mechanism must be able to read the network packets in an unstructured format from several sources (big data settings). The feature extraction mechanism must be able to structure the captured data and extract features in a distributed fashion. The Dataset used in our project has the Canadian Institute for Cybersecurity NSL dataset 2021.

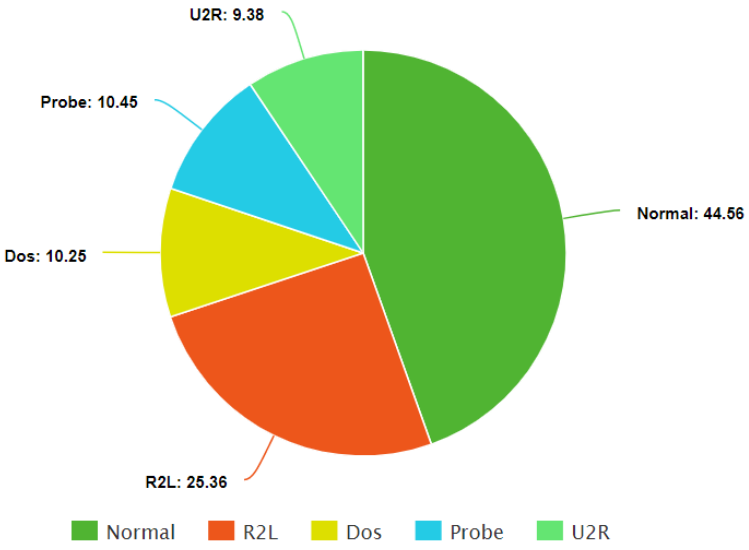

Figure 2 Multi-Class Labels Distribution

Dataset

The well organised NSL data set has two significant flaws that have a significant impact on the performance of tested systems.The large quantity of redundant data causes learning algorithms to be biased towards frequent records, preventing them from learning fewer records, which are normally more detrimental to networks such as U2R and R2L assaults. Furthermore, the presence of these repeated records in the test set frequently leads evaluation results to be skewed by approaches with higher detection rates on frequent records.

Experimental Setup

On a Windows 10 PC, all of the activities are completed with Python and the scikit-learn and TensorFlow libraries. The test computer has an Intel(R) Core i6 CPU runningthe processor at 1.8GHz and32 GB RM, 8 GPU RM, 1 GPU, 100 GB HDD.

Results

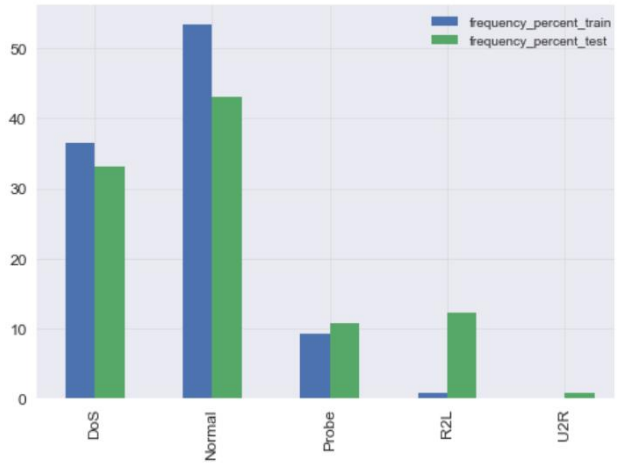

Figure 3 Frequency percentage of Train and Test Data

Figure 4 represents the Multi-class classificationof K-Nearest Neighbours how far the classification has been performed by intrusion detection. Similarly, Figures 5 and 6 depict the Support vector machines, Random forest multi-classification.

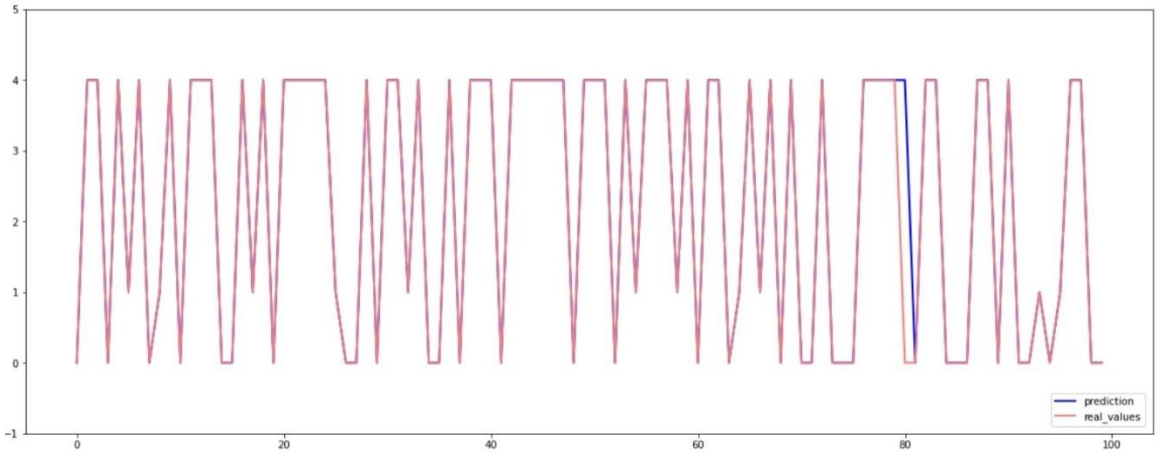

Figure 4 K-Nearest Neighbors Multi-Classification

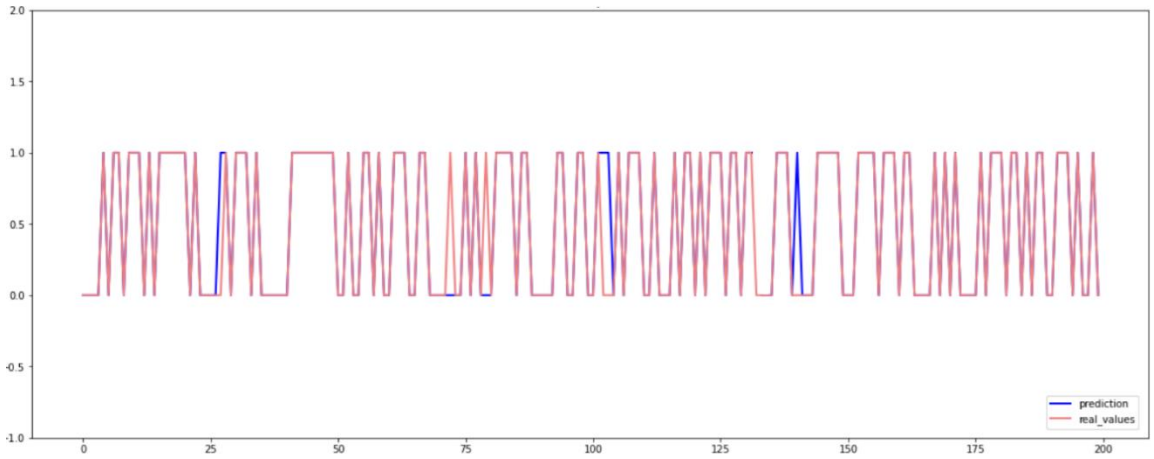

Figure 5Support Vector Machines Multi-Classification

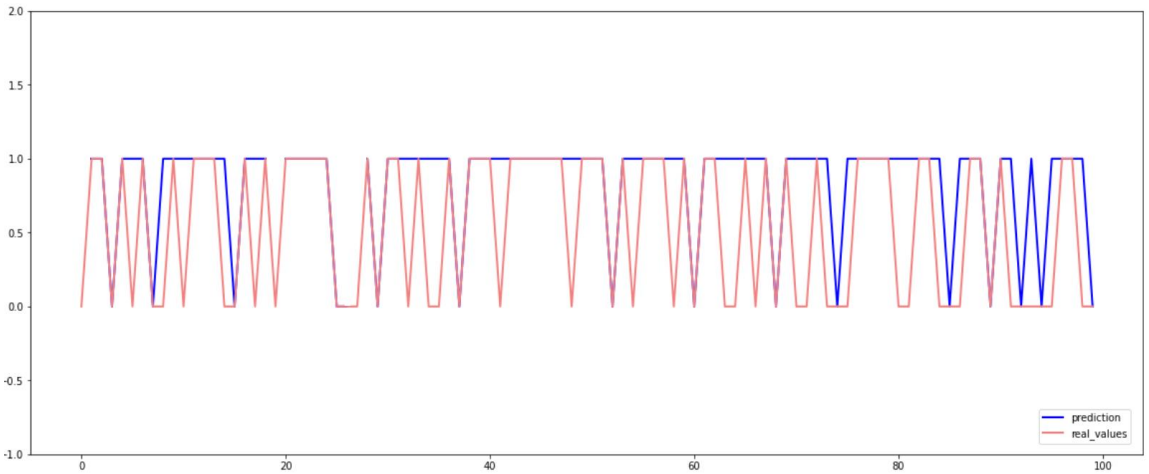

Figure 6Random Forest Multi-Classification

A receiver operating characteristic curve (ROC curve) is a graph that shows how well a machine learning classification model performs across all categorization levels.

This curve plots two parameters:

a.) True Positive Rate has calculared has following Equation

$$TPR = \frac{TP}{TP + FN}$$

b.) similarly, False Positive Rate has computed by following Equation

$$FPR = \frac{FP}{FP + TN}$$

True Positive Rate and False Positive Rate at various categorization criteria are plotted on a ROC curve. As the classification threshold is lowered, more items are classified as positive, increasing both False Positives and True Positives. A typical ROC curve for different machine learning is depicted in the diagram below Figure 7, 8 and 9.

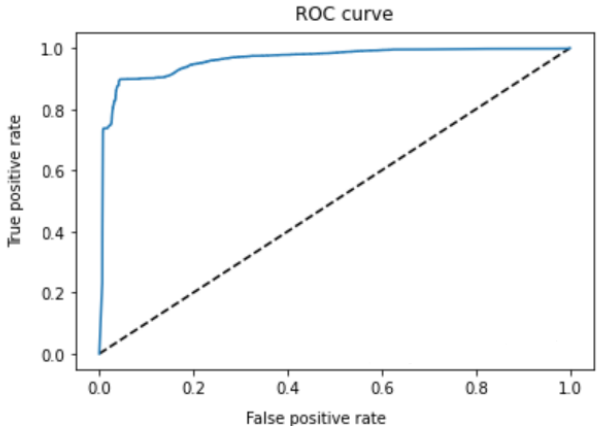

Figure 7 ROC Curve Sample 1

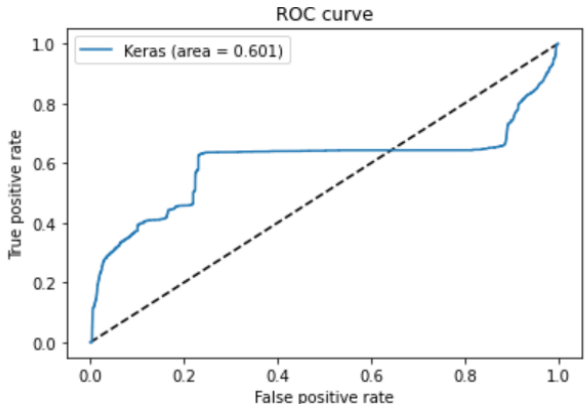

Figure 7 ROC Curve Sample 2

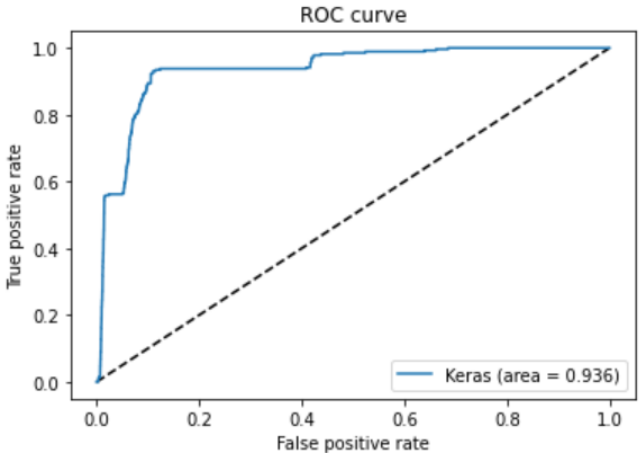

Figure 7 ROC Curve Sample 3

Discussion

The difficulty of network traffic fluctuations over time has been overlooked by suggested ML-based intrusion detection techniques. This created dataset represents a significant step forward in the accurate evaluation of machine learning-based intrusion detection techniques. To our knowledge, it is the first collection of genuine network traffic that has been previously classified, made public, and contains several years of network traffic behaviour. Even when feature selection is made, current ML-based intrusion detection systems are unable to cope with the dynamic behaviour of network traffic, as demonstrated by our constructed dataset. Within weeks of the training period, current techniques lose a considerable amount of accuracy. As a result, ML-based schemes must be updated regularly, making their application in real-world settings more difficult.

Conclusion

We have used the user NSL dataset of which contains parameters different of a keyboard, Mouse, applications running, processor usage, etc had developed a statistical engine that applies logistic regression and Statistical mean on different user's datasets and test cases with different features. To continue with the work we assign an expert. Now, experts know normal user behaviour hence, the expert provides rules in machine learning Intrusion Detection Engine PIDE. Here, we have used machine learning techniques to provide rules. The machine learning-based detection collects various data for possible attacks to identify authorized and unauthorized activities. models are to be defined in such a way that only doubtful activities are noticed without disturbing authorized users.

References

1. Elmer C. Castillano, Rose ann L. Ugbinada and Sergio R. Canoy, Jr., “Secure Domination in the Join of Graphs”, Applied Mathematical Sciences, vol. 8, no. 105, 5203 – 5211, 2014.
2. E. Sampathkumar and H. B. Walikar, “The connected domination number of a graph”, Journal of Mathematical and Physical Sciences, vol. 13, no. 6, 607 - 613, 1979.
3. Amerkhan G. Cabaro, Sergio S. Canoy, Jr. And Imelda S. Aniversario, “Secure Connected Domination in a Graph”, International Journal of Mathematical Analysis, vol. 8, no. 42, 2065 –2075, 2014.
4. Arivudainambi D., Varun Kumar K.A, Vinoth Kumar R., &Visu P., (2020). Ransomware Traffic Classification Using Deep Learning Models: Ransomware Traffic Classification. International Journal of Web Portals (IJWP), 12(1), 1-11. http://doi.org/10.4018/IJWP.2020010101
5. Arivudainambi, D., Varun Kumar, K.A., and Satapathy, Suresh Chandra. ‘Correlation Based Malicious Traffic Analysis System’. 1 Jan. 2021: 195 – 200. (4700 words)
6. T. W. Haynes, S. T. Hedetniemi and P. J. Slater, Fundamentals of Domination in Graphs, Marcel Dekker New York, 1998.

7. G. Zaki et al., "The Utility of Cloud Computing in Analyzing GPU-Accelerated Deformable Image Registration of CT and CBCT Images in Head and Neck Cancer Radiation Therapy," in IEEE Journal of Translational Engineering in Health and Medicine, vol. 4, pp. 1-11, 2016, Art no. 4300311, doi: 10.1109/JTEHM.2016.2597838.
8. J. Baliga, R. W. A. Ayre, K. Hinton and R. S. Tucker, "Green Cloud Computing: Balancing Energy in Processing, Storage, and Transport," in Proceedings of the IEEE, vol. 99, no. 1, pp. 149-167, Jan. 2011, doi: 10.1109/JPROC.2010.2060451.
9. T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta and D. Sabella, "On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration," in IEEE Communications Surveys & Tutorials, vol. 19, no. 3, pp. 1657-1681, thirdquarter 2017, doi: 10.1109/COMST.2017.2705720.
10. K. Mershad, H. Artail, M. A. R. Saghir, H. Hajj and M. Awad, "A Study of the Performance of a Cloud Datacenter Server," in IEEE Transactions on Cloud Computing, vol. 5, no. 4, pp. 590-603, 1 Oct.-Dec. 2017, doi: 10.1109/TCC.2015.2415803.
11. H. Yan et al., "Cost-Efficient Consolidating Service for Aliyun's Cloud-Scale Computing," in IEEE Transactions on Services Computing, vol. 12, no. 1, pp. 117-130, 1 Jan.-Feb. 2019, doi: 10.1109/TSC.2016.2612186.
12. C. Xia et al., "Workflow-Based Service Selection under Multi-constraints," 2015 IEEE International Conference on Services Computing, 2015, pp. 332-339, doi: 10.1109/SCC.2015.53.
13. L. A. Tawalbeh and W. Bakhader, "A Mobile Cloud System for Different Useful Applications," 2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW), 2016, pp. 295-298, doi: 10.1109/W-FiCloud.2016.66.
14. Chuan-Yen Chiang, Yen-Lin Chen, Kun-CingKe and Shyan-Ming Yuan, "Real-time pedestrian detection technique for embedded driver assistance systems," 2015 IEEE International Conference on Consumer Electronics (ICCE), 2015, pp. 206-207, doi: 10.1109/ICCE.2015.7066383.
15. A. C. F. Petri and D. F. Silva, "Towards logical association rule mining on ontology-based semantic trajectories," 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), 2020, pp. 586-591, doi: 10.1109/ICMLA51294.2020.00098.
16. F. Tian, M. Zhang, Z. Wu, X. Zhu, P. Diao and J. Wang, "Synthesis and dielectric property of polyimide/MWNTs nanocomposite films," 2009 IEEE 9th International Conference on the Properties and Applications of Dielectric Materials, 2009, pp. 829-832, doi: 10.1109/ICPADM.2009.5252196.
17. M. Bahrami and M. Singhal, "A Light-Weight Permutation Based Method for Data Privacy in Mobile Cloud Computing," 2015 3rd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, 2015, pp. 189-198, doi: 10.1109/MobileCloud.2015.36.
18. Arivudainambi, D., K.A, V. & Sibi Chakkaravarthy, S. LION IDS: A meta-heuristics approach to detect DDoS attacks against Software-Defined Networks. Neural Computing & Applications 31, 1491–1501 (2019). https://doi.org/10.1007/s00521-018-3383-7
19. Arivudainambi D., Varun Kumar K.A., Sibi Chakkaravarthy S., Visu P., Malware traffic classification using principal component analysis and artificial neural network for extreme surveillance, Computer Communications, Volume 147, 2019, Pages 50-57,
20. N. Bansal and M. Dutta, "Performance evaluation of task scheduling with priority and non-priority in cloud computing," 2014 IEEE International Conference on Computational Intelligence and Computing Research, 2014, pp. 1-4, doi: 10.1109/ICCIC.2014.7238289.
21. T. Mengistu, A. Alahmadi, A. Albuali, Y. Alsenani and D. Che, "A "No Data Center" Solution to Cloud Computing," 2017 IEEE 10th International Conference on Cloud Computing (CLOUD), 2017, pp. 714-717, doi: 10.1109/CLOUD.2017.99.
22. C. Zhang, R. Green and M. Alam, "Reliability and Utilization Evaluation of a Cloud Computing System Allowing Partial Failures," 2014 IEEE 7th International Conference on Cloud Computing, 2014, pp. 936-937, doi: 10.1109/CLOUD.2014.131.
23. M. Bahrami and M. Singhal, "A dynamic cloud computing platform for eHealth systems," 2015 17th International Conference on E-health Networking, Application & Services (HealthCom), 2015, pp. 435-438, doi: 10.1109/HealthCom.2015.7454539.