IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

DATA MISMATCH AND ERROR DETECTION USING CLOUD

M. Preetha[1], Ashwin Kumar P J[2], Dinesh D[3],Harish Suriya S[4]
[1]AssistantProfessor,[2,3,4] Students,
Department of Computer Science and Engineering,
PERI Institute of Technology,
Chennai, India.

Abstract*:* In today’s interconnected world, ensuring the security of cloud-based systems is paramount. Traditional intrusion detection systems (IDS) often struggle to keep pace with the evolving landscape of cyber threats. This project proposes a novel approach to enhancing cloud security through the integration of machine learning techniques into intrusion detection. By harnessing the power of machine learning algorithms, our system can adapt and learn from patterns in network traffic data, enabling it to detect anomalous behavior indicative of potential intrusions. Leveraging the scalability and flexibility of cloud computing, our solution offers real-time monitoring and analysis of network traffic across distributed cloud environments. Key features of our cloud-based intrusion detection system include automated threat detection, rapid response mechanisms, and customizable alerting capabilities. Through continuous learning and refinement, the system improves its detection accuracy over time, bolstering the resilience of cloud infra structures against emerging cyber threats. In summary, our project presents a proactive and adaptive approach to safeguarding cloud-based systems, combining the strengths of machine learning and cloud computing to create a robust Defense mechanism against intrusions. Protecting cloud-based systems from unauthorized access, data breaches, and other malicious activities is a critical concern for organizations worldwide. Traditional intrusion detection systems (IDS) often fall short in effectively identifying and mitigating these threats in dynamic cloud environments .To address this gap, our project focuses on leveraging the Random Forest algorithm to enhance cloud-based intrusion detection capabilities. High Accuracy Random Forest excels in handling complex, high- dimensional data and can effectively distinguish between normal and malicious network activities with high accuracy.

I. INTRODUCTION

Cloud technologies give users more options for their service models and enable convenient on demand access to shared networks, storage, and resources. These concepts, which are utilized in the private, public, and hybrid cloud deployment models, are platform as a service (PaaS), software as a service (SaaS), and infrastructure as a service (IaaS). Based on its features, the cloud offers high-performance services, as stated by the National Institute of Standards and Technology.

Despite of more solutions given to secure cloud environments, the recent intrusion detection systems (IDSs) are affected by various significant limitations, for example, huge amounts of analyzed data ,real-time detection, data quality, and others that aim to decrease the performance of detection models.

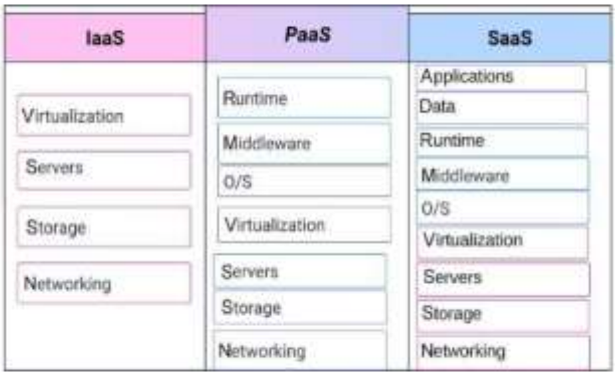


Fig.1. Cloud Models

Now a days, academic researchers show that intelligent learning methods. such as machine learning (ML), deep learning (DL), and ensemble learning are useful in various areas and are able to perform network security.

Our main goal in this research work is to propose an anomaly detection approach based on random forest (RF) bi- nary classifier and feature engineering is carried out based on a data visualization process aiming to reduce the number of used features and perform the proposed anomaly detection model. The evaluation performances of the model are implemented on NSL-KDD and BoT-IoT datasets.

Cloud deployment models are intended for different entities as needed. The public cloud is a model that intends its resources for public clients, as the name suggests. However, the private cloud is only for one entity. The hybrid cloud concept combines both private and public clouds. Community cloud is a multi-tenant platform that allows multiple companies to collaborate on the same platform if their needs and concerns are similar. The most important difference between the public cloud and the private cloud is that the private cloud is considered the most secure since it has fewer users than the public cloud. Intrusion is a kind of unauthorized activity that could pose a possible threat to the information’s confidentiality, integrity, and availability.

II. PROPOSED FRAMEWORK

In our proposed architecture, SCADA data is collected from sensors and devices deployed across the industrial environment and transmitted to the cloud infrastructure for processing. The data preprocessing module is responsible for cleansing and formatting the raw SCADA data, including tasks such as removing outliers, handling missing values, and standardizing data formats.

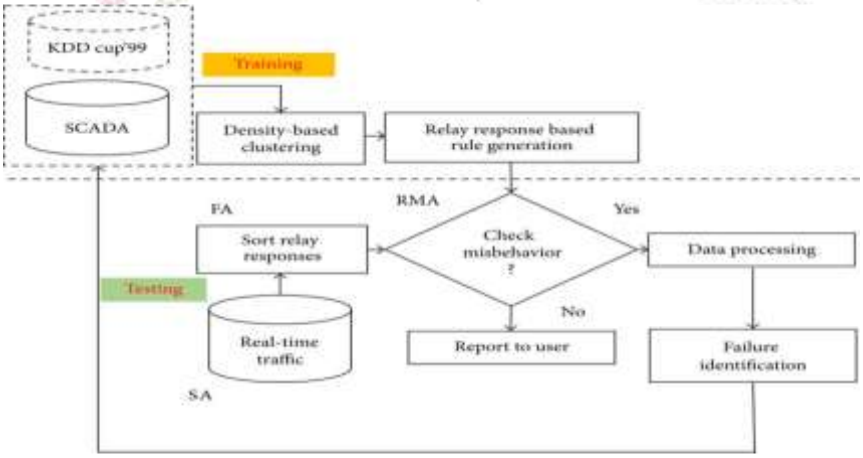


Fig.2. Architecture

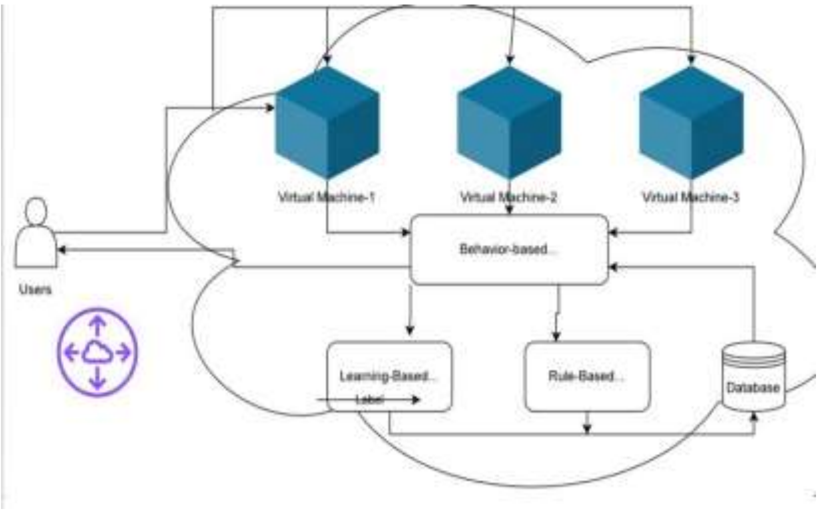
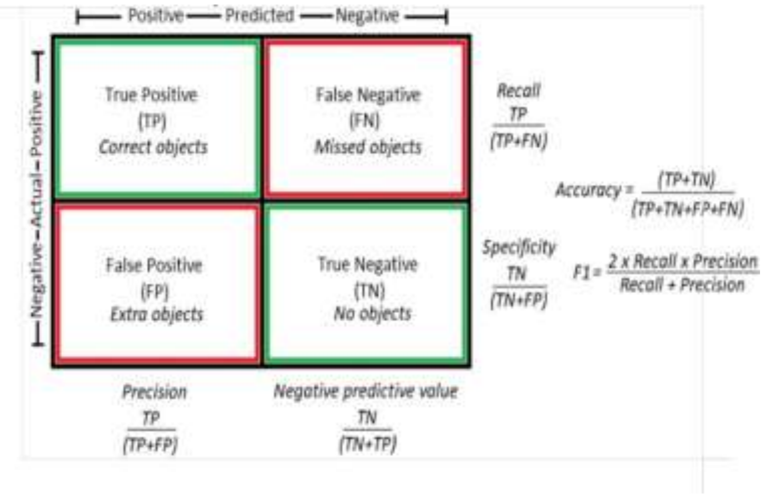


Fig.3. Virtual Machine

Subsequently, the preprocessed data is fed into a density-based clustering algorithm, such as DBSCAN (Density-Based Spatial Clustering of Applications with Noise), to identify clusters and anomalies within the dataset. The error detection and correction module then analyzes the clustered data to flag instances of data mismatches and errors, enabling timely intervention by operators or automated systems. Finally, the reporting and visualization interface provides actionable insights and alerts to operators, facilitating decision-making and proactive maintenance efforts.

SCADA (Supervisory Control and Data Acquisition) systems play a crucial role in various industries, including manufacturing, energy, and transportation, by enabling real-time monitoring and control of critical processes. However, these systems are prone to data mismatches and errors due to factors such as sensor malfunctions, communication issues, and data corruption. These discrepancies can lead to inefficiencies in production, equipment failures, and even safety hazards for personnel and the environment. Recognizing the importance of addressing these challenges, our proposed system leverages cloud computing infrastructure and advanced data analysis techniques to automatically detect and mitigate data mismatches and errors in SCADA systems.

Our proposed system addresses the critical challenge of detecting data mismatches and errors in industrial processes by leveraging density-based clustering techniques in a cloud computing environment. With the proliferation of sensor net- works and the increasing volume of data generated in industrial settings, traditional error detection methods have become inadequate, necessitating more sophisticated approaches. By harnessing the scalability and processing power of cloud infrastructure, our system offers real-time analysis of large-scale SCADA data streams.



The core of our approach lies in employing density-based clustering algorithms, such as DBSCAN (Density-Based Spatial Clustering of Applications with Noise),to identify clusters of data points that represent normal operational patterns. Deviations from these clusters are indicative of potential mismatches or errors, enabling proactive detection and mitigation. The cloud-based architecture ensures seamless integration with existing SCADA systems, facilitating rapid deployment and scalability. Moreover, by automating the error detection process, our system reduces the reliance on manual intervention, leading to improved operational efficiency and reduced down- time. Overall, our proposed system represents a promising solution for enhancing the reliability and integrity of industrial data streams in an era of digital transformation.”

III. ALGORITHM

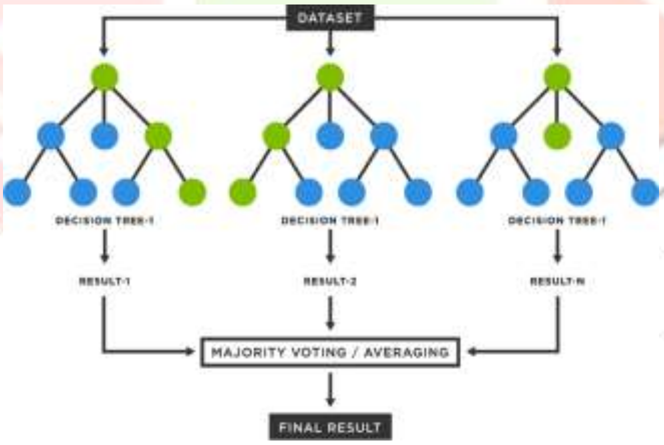
Random Forest is a powerful ensemble learning algorithm that is widely used for classification and regression tasks in machine learning. It belongs to the family of decision tree- based algorithms and is known for its robustness, scalability, and ability to handle large datasets with high-dimensional feature spaces. Here’s an overview of how the Random Forest algorithm works:

Random Forest builds multiple decision trees during training. Each decision tree is trained independently on a random subset of the training data and features. Decision trees partition the feature space into regions and make predictions by traversing the tree from the root to a leaf node.

Random Forest employs a technique called bootstrap aggregating, or bagging, to create diverse decision trees. It randomly samples the training data with replacement to create multiple subsets of data for training each decision tree. This helps in reducing over fitting and improves the generalization ability of the model.

In addition to bootstrapping, Random Forest also performs random feature selection when building each decision tree. Instead of considering all features at each split, it randomly selects a subset of features. This randomness further decor relates the trees and enhances the diversity of the ensemble.

During prediction, each decision tree in the forest independently makes a prediction. For classification tasks, the final prediction is typically determined by majority voting among the individual trees. For regression tasks ,the final prediction is often the average or median of the predictions from all trees.



Random Forest has several hyper parameters that can be tuned to optimize its performance, such as the number of trees in the forest, the maximum depth of each tree, the number of features to consider at each split, and the criterion for splitting nodes (e.g., Gini impurity or information gain).

- Random Forest is robust to over fitting and noise in the data, thanks to its ensemble approach and randomization techniques.
- Random Forest provides estimates of feature importance, which can be useful for understanding the underlying data patterns and feature engineering.

Random Forest is widely used across various domains, including but not limited to:

- Predictive maintenance in manufacturing
- Disease diagnosis in healthcare
- Fraud detection in finance
- Customer churn prediction in marketing
- Image classification and object detection in computer vision

The Random Forest algorithm, a prominent ensemble learning technique, is extensively employed in both classification and regression tasks within the domain of machine learning. Renowned for its versatility and robustness, Random Forest operates by constructing an ensemble of decision trees during the training phase. Each decision tree is developed independently, utilizing a randomly selected subset of both the training data and features. This diversity a cross the ensemble mitigates the risk of over- fitting and enhances the model's generalization capabilities. During the prediction phase, each decision tree in the forest generates its individual prediction, and the final output is determined through majority voting for classification tasks or averaging for regression tasks. Moreover, Random Forest incorporates randomness not only in the data sampling but also in feature selection at each node split, further enhancing the diversity among the constituent trees. Its ability to handle high-dimensional feature spaces, mitigate over fitting, and provide insights into feature importance makes Random Forest a popular choice across various domains, including healthcare, finance, manufacturing, and marketing.

IV. Existing Framework

4.1. Data Collection

Gather the dataset containing network traffic or system logs to be analyzed for intrusion detection.

4.2. Pre-Processing (Feature Selection)

Select relevant features from the raw data and perform data cleaning (handling missing values, duplicates, normalization).

Train the SVM model using the processed data. Tune SVM hyper parameters (like kernel type, regularization) to optimize performance. Test the trained SVM model with a separate dataset to assess accuracy and speed .Calculate performance metrics like accuracy, precision, recall, and F1-score. Compare the SVM model's results with base line models to gauge its relative effectiveness. Validate results through additional tests or cross-validation.

4.3. Deployment and Monitoring

Deploy the trained SVM model in a real-time environment for continuous intrusion detection. Implement monitoring and alerting systems to detect changes in network traffic patterns and notify security personnel.

4.4. Response and Feedback

Define response protocols for handling detected threats. Establish a feedback loop to update the SVM model with new data and maintain its accuracy over time. In the realm of data mismatch and error detection, particularly within the context of cloud computing, existing systems often leverage density- based clustering algorithms to identify anomalies in large datasets. These systems typically involve the preprocessing of data collected from diverse sources, such as sensors in industrial environments or IoT devices in smart cities. Once the data is cleansed and standardized, density-based clustering algorithms, like DBSCAN (Density-Based Spatial Clustering of Applications with Noise) or OPTICS (Ordering Points To Identify the Clustering Structure),are applied to partition the dataset into clusters based on the density of data points. Anomalies, or data points that fall outside the dense regions, are flagged as potential mismatches or errors.

However, existing systems may face challenges in scalability and computational efficiency when dealing with massive datasets in the cloud. Moreover, the effectiveness of anomaly detection heavily depends on the choice of parameters and the quality of the input data. Despite these challenges, existing systems have demonstrated promising results in detecting anomalies and improving data integrity in cloud-based environments, contributing to enhanced decision-making and operational efficiency across various domains.

V. Module

Scikit-learn: Scikit-learn is a simple and efficient tool for data mining and data analysis in Python. It provides a wide range of machine learning algorithms and tools for tasks like classification, regression, clustering, and dimensionality reduction. Scikit-learn includes implementations of popular machine learning algorithms such as decision trees, support vector machines, random forests, and k-nearest neighbors.

Imbalanced-learn: It is an extension library built upon Scikit-learn, addresses the challenges posed by imbalanced datasets in machine learning tasks .Imbalanced datasets, where the distribution of classes is skewed, are prevalent in various real-world scenarios such as fraud detection, anomaly detection,andmedicaldiagnosis.Imbalanced-learnprovidesacomprehensive set of algorithms and tools specifically designed to tackle this issue and improve the performance of machine learning models on imbalanced data.

Pandas: It is a powerful data manipulation and analysis library in Python, revolutionizes the way data is handled and processed in data science and machine learning work- flows. With its intuitive and flexible data structures, primarily Data Frame and Series objects, P and as provides a rich set of functionalities for loading, cleaning, transforming, and analyzing structured data.

NumPy: It is a fundamental library for numerical computing in Python, plays a pivotal role in enabling efficient manipulation and computation with multi-dimensional arrays. Its versatility and performance make it a cornerstone of the Python scientific computing eco system, powering a wide range of applications in data analysis, machine learning, signal processing, and more.

Pcapy: Pcapy is a Python library that provides a convenient interface for capturing network packets and performing network analysis tasks. Leveraging the capabilities of libpcap, pcapy enables users to capture packets from network interfaces in real-time or read packets from saved packet capture (pcap) files. This functionality makes pcapy invaluable for network monitoring, intrusion detection, and forensic analysis applications. With pcapy, users can filter captured packets based on various criteria, such as protocol type, source or destination IP address, and port number, enabling targeted analysis of network traffic.

Scapy: Scapy is a powerful Python library used for crafting, sending, and analyzing network packets. It provides a wide range of functionalities for network protocol development, testing, and security auditing. With Scapy, users can create custom packets from scratch or modify existing ones, making it a valuable tool for network research, penetration testing, and protocol analysis.

Joblib: Joblib is a Python library designed to provide lightweight pipelining in Python, optimized for computational and memory-intensive tasks. It is particularly useful for parallelizing code execution, caching function results, and saving and loading Python objects efficiently.

Seaborn: Seaborn is a Python data visualization library built on top of Matplotlib that provides a high-level interface for creating attractive and informative statistical graphics. It is designed to work seamlessly with Pandas data structuresandisparticularlyusefulforvisualizingcomplexdatasetswith minimal code.

Django: Django is a high-level web framework written in Python that encourages rapid development and clean, pragmatic design. It follows the ”batteries-included” philosophy, providing developers with everything they need to build web applications efficiently and securely.

VI. Methodology

The methodology for data mismatch and error detection using density-based clustering in the cloud typically involves several key steps. Initially, raw data collected from disparate sources is preprocessed to remove noise, handle missing values, and standardize the format. Raw data streams from SCADA systems, which capture real-time operational metrics from industrial processes, are collected and aggregated. This data undergoes preprocessing to address inconsistencies, outliers, and missing values, ensuring the data’s quality and uniformity for subsequent analysis. Subsequently, the preprocessed data is fed into a density-based clustering algorithm, such as DBSCAN (Density-Based Spatial Clustering of Applications with Noise), within the cloud environment.

This clustering algorithm partitions the data into clusters based on the density of observations, thereby identifying nor- mal operational patterns within the SCADA data. Anomalies, or data points lying outside these dense regions, are flagged as potential mismatches or errors. To optimize the clustering process, parameters such as minimum points and epsilon radius are adjusted, balancing the algorithm’s sensitivity to anomalies and its tolerance for noise. Leveraging the scalability and parallel processing capabilities of the cloud, this methodology enables the efficient analysis of large volumes of SCADA data in real-time.

Techniques such as cross-validation and holdout validation are employed to validate the effectiveness of the methodology and refine parameters as necessary. Overall, the methodology for data mismatch and error detection using SCADA and density-based clustering in the cloud offers a robust framework for enhancing operational efficiency and ensuring the integrity of industrial data.

This preprocessing step is crucial for ensuring the quality and consistency of the data before applying clustering algorithms. Subsequently, density-based clustering algorithms such as DBSCAN or OPTICS are employed to partition the preprocessed dataset in to clusters based on the density of data points. These algorithms identify dense regions in the data space, which are considered normal operational patterns, and classify data points outside these regions as anomalies or potential errors.

Parameters such as minimum points and epsilon radius are tuned to optimize the clustering process, balancing between sensitivity to anomalies and robustness to noise. In the cloud environment, the scalability and parallel processing capabili- ties are leveraged to handle large volumes of data efficiently.

Mechanisms for model evaluation and validation are implemented to assess the performance of the clustering algorithm in detecting anomalies. Techniques such as cross-validation or holdout validation are utilized to validate the effectiveness of the methodology and fine-tune parameters if necessary. Overall, the methodology for data mismatch and error detection using density-based clustering in the cloud is iterative and involves a combination of data preprocessing, algorithm selection, parameter tuning, and evaluation to ensure accurate and reliable detection of anomalies in the data.

VII. Conclusion

In this research, we offer a method for cloud security intrusion detection using RF and graphic display. Next, features engineering is done using the first, while intrusion detection and prediction are done with the second. prior to the model’s training. According to the results, the RF classifier outperforms DNN, DT, and SVM in terms of accuracy when it comes to attack type prediction and classification. By comparing the outcomes with those of other classifiers, we have illustrated the possibility of using few features. However, recall using NSL-KDD is still not good enough, thus in our next study, we will address this issue by enhancing our model with DL and ensemble learning approaches. The pursuit of accurate and efficient data mismatch and error detection is paramount in various domains, ranging from industrial processes to financial transactions and healthcare systems. Leveraging advanced techniques such as density- based clustering coupled with cloud computing infra structure, organizations can effectively identify anomalies and discrepancies within their datasets. Through meticulous preprocessing, feature engineering, and parameter optimization, these methodologies enable the detection of subtle deviations indicative of errors or mismatches, ensuring data integrity and operational efficiency. Addition- ally, the scalability and adaptability afforded by cloud-based solutions empower organizations to handle large volumes of data in real-time, facilitating timely intervention and decision- making. As technology continues to evolve, so too will the methodologies and tools available for data mismatch and error detection. By embracing innovations in data analytics, machine learning, and cloud computing ,organizations can stay ahead of the curve, safeguarding their data assets and optimizing their processes for sustained success.

Reference

[1] Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, Survey of intrusion detection systems: Techniques, datasets and challenges, Cybersecurity, vol. 2, p. 20, 2019.

[2] Guezzaz, A. Asimi, Y. Asimi, Z. Tbatou, and Y. Sadqi, A global intrusion detection system using PcapSockS sniffer and multilayer perceptron classifier, International Journal of Network Security, vol. 21, no. 3, pp. 438–450, 2019.

[3] Guezzaz, S. Benkirane, M. Azrour, and S. Khurram, A reliable network intrusion detection approach using decision tree with enhanced data quality, Security and Communication Networks, vol. 2021, p. 1230593, 2021.

[4] A. Tama and K. H. Rhee, HFSTE: Hybrid feature selections and tree-based classifiers ensemble for intrusion detectionsystem,IEICETrans.Inf.Syst.,vol.E100.D,no. 8, pp. 1729–1737, 2017.

[5] M. Azrour, J. Mabrouki, G. Fattah, A. Guezzaz, and F. Aziz, Machine learning algorithms for efficient water quality prediction,ModelingEarthSystemsandEnvironment,vol.8,pp.2793–2801,2022.

[6] M. Azrour, Y. Farhaoui, M. Ouanan, and A. Guezzaz, SPIT detection in telephony over IP using K-means algorithm, Procedia Computer Science, vol. 148, pp. 542–551, 2019.

[7] M. Azrour, M. Ouanan, Y. Farhaoui, and A. Guezzaz, Security analysis of Ye et al. authentication protocol for internet of things, in Proc. International Conference on Big Data and Smart Digital Environment, Casablanca, Morocco, 2018, pp. 67–74.

[8] M. Azrour, J. Mabrouki, A. Guezzaz, and A. Kanwal, Internet of things security: Challenges and key issues, Security and Communication Networks, vol. 2021, p. 5533843, 2021. Ayei E. Ibor (2022) A

Hybrid Mitigation Technique for Ma- licious Network Traffic based on Active Response HYBRITQ- 4(J48, Boyer Moore, K- NN) High detection rate and low falsepositive rate.

[9] AnnkitaPatel,Risha Tiwari (2022) Bagging Ensemble Tech- nique for intrusion Detection System Bagging & Boosting: SVM & Decision Tree Combination of two algorithms is better than other ensemble technique.

[10] Yogita B. Bhavsar&Kalyani C. Waghmare (2022) Intru- sion Detection System Using Data Mining Technique Support Vector Machine Data Mining , SVM, Detection rate is in- creased & False positive rate is decreased.

[11] Rowayda A. Sadek, M. Sami Soliman& Hagar S. Elsayed (2022) Effective Anomaly Intrusion Detection System based on Neural Network with Indicator Variable and Rough set Reduction NNIV-RS High detection rate and low false positive rate.

[12] Ahemd A Elngar, Dowalt, Fayed (2021) A Real Time Anomaly Intrusion Detection System with High Accuracy PSO-Discritize- HNB High detection accuracy and speed up the time.

[13] HeshamAltwaijry, Saeed Algarny (2021) Bayesian based Intrusion Detection System Bayesian Probability Achieved better detection rate with low threshold value.

[14] Long short term memory recurrent neural network classifier for intrusion detection developed by J. Kim, J. Kim, H. L.
T. Thu, and H. Kim was published in Proceedings of the 2016 International Conference on Plateform Technology and Service, Jeju, Republic of Korea, 2016, pp. 1–5.

[15] J. Zhang, Multimedia Tools and Applications, vol. 78, pp. 30923–30942, 2019; anomaly detection and ranking of the cloud computing platform by multi-view learning.

[16] Securing cloud data: A machine learning based data clas- sification technique for cloud computing, F. B. Ahmad, A. Nawaz, T. Ali, A. A. Kiani, and G. Mustafa, http://doi.org/ 10.21203/rs.3.rs-1315357/v1, 2022.