*Anomaly Detection in Networks Using Machine Learning*

Shelly Shiju George

*Master of Computer Applications*

*Amal Jyothi College Of Engineering*

Kanjirappally, India

shellyshijugeorge@amaljyothi.ac.in

Rugma S

*Master of Computer Applications*

*Amal Jyothi College Of Engineering*

Kanjirappally, India

rugmas2024b@mca.ajce.in

*Abstract*— The passage discusses the increasing number of attacks on the internet despite the rapid growth in its usage over the past two decades. It highlights the limitations of signature-based methods in preventing attacks, particularly zero-day attacks, which are not known or accounted for in existing security measures. To address this, the passage proposes the use of anomaly-based approaches, which have the potential to detect zero-day attacks. The study aims to detect network anomalies using machine learning methods, with a focus on the CICIDS2023 dataset due to its relevance and diverse range of attack types. Feature selection was performed using the Random Forest Regressor algorithm. Seven different machine learning algorithms were then applied in the detection process, resulting in high performance rates. The success rates achieved by each algorithm are listed as follows: Naive Bayes (NB) - 86%, Quadratic Discriminant Analysis (QDA) - 86%, Random Forest (RF) - 94%, Iterative Dichotomiser 3 (ID3) - 95%, Adaptive Boosting (AdaBoost) - 94%, Multi-Layer Perceptron (MLP) - 83%, and K Nearest Neighbors (KNN) - 97%.

Keywords—Machine Learning, Network Security, Network Anomaly Detection, Naive Bayes, QDA, Random Forest, ID3, AdaBoost, MLP, KNN

# Introduction

The passage discusses two primary methods for detecting internet attacks: signature-based and anomaly-based approaches. It notes the increasing frequency of attacks and the challenges faced by signature-based methods, which rely on databases that require constant updating and are susceptible to zero-day attacks. Anomaly-based detection, on the other hand, focuses on identifying unusual network behaviors and has shown effectiveness against zero-day attacks.The prevalence of encrypted internet traffic, particularly through SSL/TLS protocols, poses challenges for signature-based methods since they cannot inspect message content. However, anomaly-based approaches analyze general properties of data, such as size and connection time, enabling effective analysis of encrypted protocols without needing to view message content. This advantage has led to increased adoption of anomaly-based detection methods for preventing network attacks.The study aims to contribute to the literature by developing a system that utilizes machine learning methods to quickly and effectively detect network anomalies.

# Literature Review

Engly et al. [1] presented a study at the International Conference on Network and System Security, focusing on enhancing intrusion detection systems through anomaly-based techniques with combined imbalance correction and feature selection methods.

Hussain et al. [2] proposed a comprehensive framework for classifying denial of service attacks, aiming to systematically categorize and mitigate various forms of service disruptions. Their work was presented at the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications.

Vincent et al. [3] introduced stacked denoising autoencoders for learning useful representations in deep networks, revolutionizing deep learning by enabling the extraction of meaningful features from noisy data. Their research was published in the Journal of Machine Learning Research.

Al-Qatf et al. [4] developed a novel deep learning approach for network intrusion detection, combining sparse autoencoders with SVM for enhanced accuracy. Their method, outlined in IEEE Access, contributed to more robust cybersecurity measures.

Masci et al. [5] introduced stacked convolutional autoencoders for hierarchical feature extraction, revolutionizing image processing and pattern recognition by enabling automatic extraction of complex features from raw data. Their work was presented at the International Conference on Artificial Neural Networks.

Park et al. [6] implemented network intrusion detection using stacked denoising autoencoders, demonstrating the effectiveness of deep learning techniques in identifying and mitigating network security threats. Their approach was outlined in Advances in Science Letters.

Chen et al. [7] proposed an autoencoder-based network anomaly detection method at the Wireless Telecommunications Symposium, introducing an innovative approach for detecting abnormalities in network traffic, particularly relevant for wireless communication systems.

Shone et al. [8] presented a deep learning approach to network intrusion detection in IEEE Transactions on Emerging Topics in Computational Intelligence, showcasing the power of deep neural networks in accurately identifying and mitigating network security breaches, with potential implications for future research in cybersecurity.

Lakhina et al. [9] discussed diagnosing network-wide traffic anomalies in ACM SIGCOMM Computer Communication Review, providing insights into understanding and diagnosing anomalous network behavior, which can facilitate more effective network management and security protocols.

# Methodology

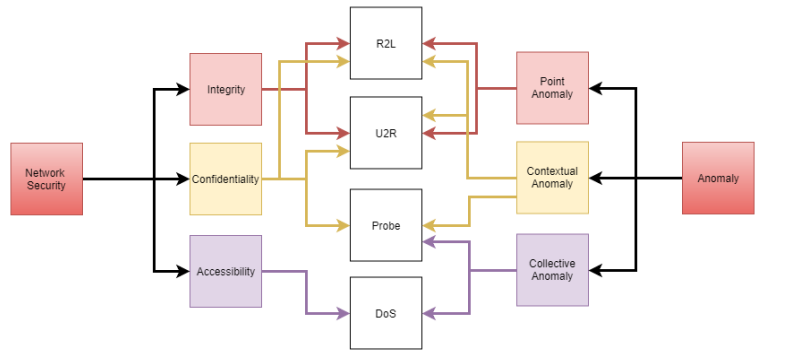
The study introduces a methodology for detecting network anomalies using machine learning techniques to tackle the increasing threat landscape of internet-based attacks. It highlights the limitations of signature-based methods in combating zero-day attacks and advocates for the adoption of an anomaly-based approach, which has shown promise in detecting such attacks. The methodology utilizes contemporary datasets with diverse attack scenarios for robust evaluation of anomaly detection techniques. The first step involves feature selection using the Random Forest Regressor algorithm to identify the most discriminative features for anomaly detection. Subsequently, seven different machine learning algorithms are applied in the detection phase: Naive Bayes, Quadratic Discriminant Analysis (QDA), Random Forest, ID3, AdaBoost, Multi-Layer Perceptron (MLP), and K Nearest Neighbors (KNN). These algorithms are chosen for their versatility and established effectiveness in anomaly detection tasks. Through rigorous evaluation, the methodology achieves high performance rates, demonstrating the efficacy of machine learning in network anomaly\_detection.

Fig 1. The relationship between network anomalies and networks attacks.

## About dataset

The CICIDS 2023 dataset, curated by the Canadian Institute for Cybersecurity at the University of New Brunswick, provides a real-world glimpse into network traffic and security incidents. Covering a 5-day period in July 2023, it captures a wide range of network activities, including normal traffic and various types of attacks such as FTP-Patator, SSH-Patator, DoS attacks, web attacks (like brute force, XSS, and SQL injection), infiltration, botnets, DDoS, and port scans. One of its key strengths is its detailed labeling and feature extraction. With 85 features extracted from the raw network packet capture (pcap) files, researchers have access to both processed data in CSV format and the original pcap files, allowing for thorough analysis and experimentation. Additionally, the dataset includes a diverse range of protocols, such as HTTPS, FTP, HTTP, SSH, and email protocols, offering researchers a rich source of information for developing and evaluating machine learning-based anomaly detection and intrusion detection systems. Combined with the up-to-date assortment of attacks sourced from recent cybersecurity reports, the CICIDS 2023 dataset is a valuable resource for advancing network security research and enhancing cybersecurity measures.

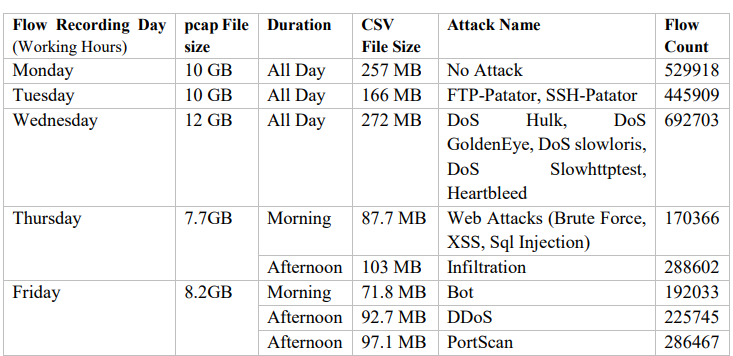


Fig 2. Details of CICIDS2017 Dataset

## Data processing

Data preprocessing workflow tailored for network traffic data stored in CSV files. Beginning with the initialization of essential variables and the import of necessary libraries such as pandas and sklearn, the script systematically addresses various preprocessing challenges. It systematically traverses through each CSV file in the specified list, diligently removing headers and rectifying inconsistencies like incomplete streams and missing values. By replacing non-numeric entries like 'inf', 'Infinity', and 'NaN' with suitable numeric equivalents or zeros, data integrity is preserved. Categorical features undergo transformation into numerical representations via label encoding techniques, while redundant columns are pruned to streamline the dataset. Subsequently, the processed data is amalgamated into a unified 'all\_data.csv' file, facilitating seamless downstream analysis. Throughout the process, informative feedback on the completion of each file's preprocessing and the total execution time ensures transparency and efficiency.

In essence, the script exemplifies a meticulous approach to preparing raw network traffic data for subsequent analysis or machine learning endeavors. By adhering to best practices in data cleansing, standardization, and consolidation, it lays a solid foundation for uncovering insights into network behavior and identifying potential anomalies or security threats. This systematic preprocessing pipeline not only enhances the interpretability and reliability of the data but also empowers researchers and analysts to extract meaningful intelligence from complex network datasets with confidence.

## Model training

A model training process for cybersecurity tasks, particularly focused on feature importance analysis using Random Forest regression. It begins by loading the preprocessed data from 'all\_data.csv', converting labels into binary values (1 for normal and 0 for attack instances), and preparing the feature matrix X and target vector y. The Random Forest Regressor is then employed to compute feature importances, quantifying the significance of each feature in predicting cyber threats.

Subsequently, the code visualizes the top 20 features based on their importance scores, generating bar plots for each dataset. Additionally, it saves the plots as PDF files in a folder named 'feature\_pics'. The feature importance lists are also printed, providing insights into the most influential features for distinguishing between normal and attack instances. This comprehensive analysis aids in understanding the underlying patterns and characteristics of cyber threats present in the dataset, facilitating informed decision-making in model training and cybersecurity strategy development. Overall, the code snippet offers a systematic approach to feature selection and model training, crucial steps in building effective cybersecurity detection systems.

# Implementation

## Import necessary libraries

To facilitate model evaluation and performance analysis, the script imports necessary libraries including scikit-learn, pandas, matplotlib, numpy, and time. These libraries provide essential functionalities for building, training, and evaluating machine learning models for cybersecurity tasks. Additionally, modules for specific algorithms such as Quadratic Discriminant Analysis (QDA), Extra Trees Classifier, Random Forest Classifier, AdaBoost Classifier, Gaussian Naive Bayes, K-Nearest Neighbors Classifier, Multi-Layer Perceptron Classifier, and Decision Tree Classifier are imported to experiment with various classification approaches. Furthermore, metrics modules like f1\_score, recall\_score, precision\_score, average\_precision\_score, and confusion\_matrix are imported to enable comprehensive evaluation of model performance in terms of accuracy, precision, recall, and other relevant metrics. Importing the warnings module helps handle potential runtime warnings, ensuring smooth execution of the code and providing a robust framework for model evaluation and performance analysis in cybersecurity tasks.

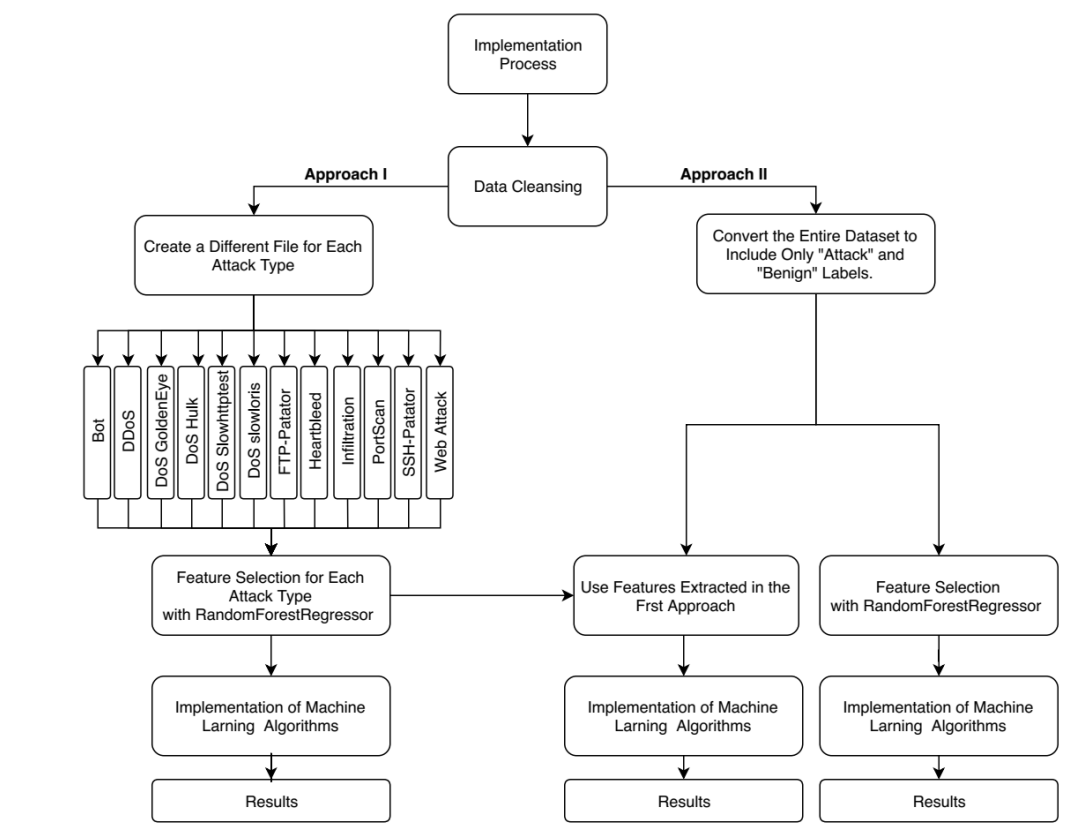


Fig 3. The Implementation Process

## Model Training

Methodical approach to preprocessing network traffic data for cybersecurity analytics, followed by model training. Initially, it standardizes numerical features, encodes categorical attributes, and consolidates multiple CSV files into 'all\_data.csv'. In model training, the dataset is divided into features (X) and labels (y), with X representing network traffic attributes and y indicating normal or attack instances. Various machine learning algorithms like RandomForestClassifier and DecisionTreeClassifier are utilized for model training. The effectiveness of trained models is evaluated using performance metrics, crucial for bolstering cybersecurity resilience in contemporary digital landscapes.

## Applying Machine Learning

the code implements various machine learning algorithms such as Naive Bayes, Quadratic Discriminant Analysis (QDA), Multi-Layer Perceptron (MLP), Random Forest, Decision Tree (ID3), AdaBoost, and k-Nearest Neighbors (KNN) to train models on preprocessed network traffic data. The performance of these models is evaluated using metrics like accuracy, precision, recall, and F1-score through repeated cross-validation iterations. Additionally, the results are visualized using box plots for further analysis. Overall, the code exemplifies a comprehensive approach to model training and evaluation for cybersecurity analytics.

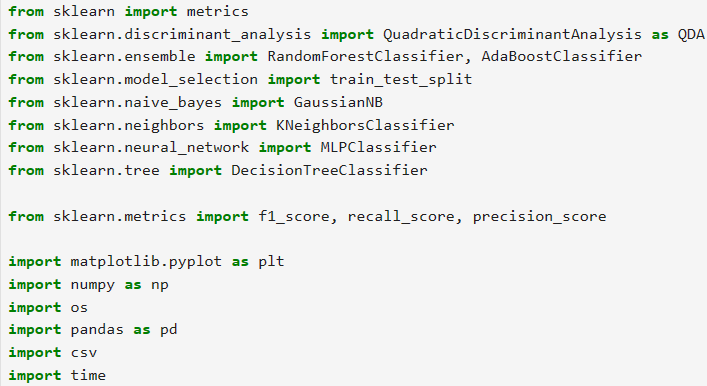


Fig 4. Machine Learning

## Model Testing

The provided code is a Streamlit application for generating attack data based on a predefined set of attacks and merging specific web attack files into a single file. The process involves reading a master dataset containing network traffic data and creating individual attack files based on attack types and a specified frequency of benign instances. The generated attack files are then merged into a single file for further analysis. The application provides real-time updates on the progress of file generation and merging. Additionally, it displays the total operation time upon completion. Overall, the code streamlines the process of generating attack data, which is essential for model testing and evaluation in cybersecurity analytics.

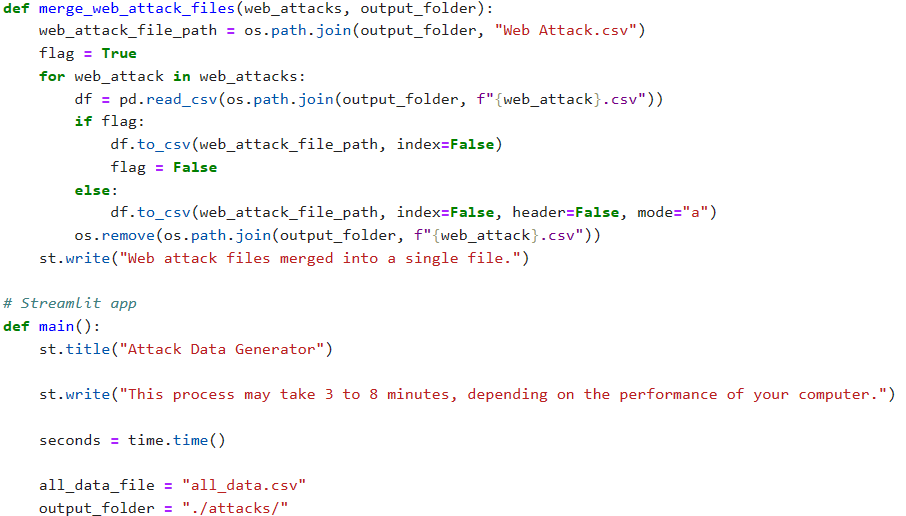


Fig 5. Preprocessing of CSVs file

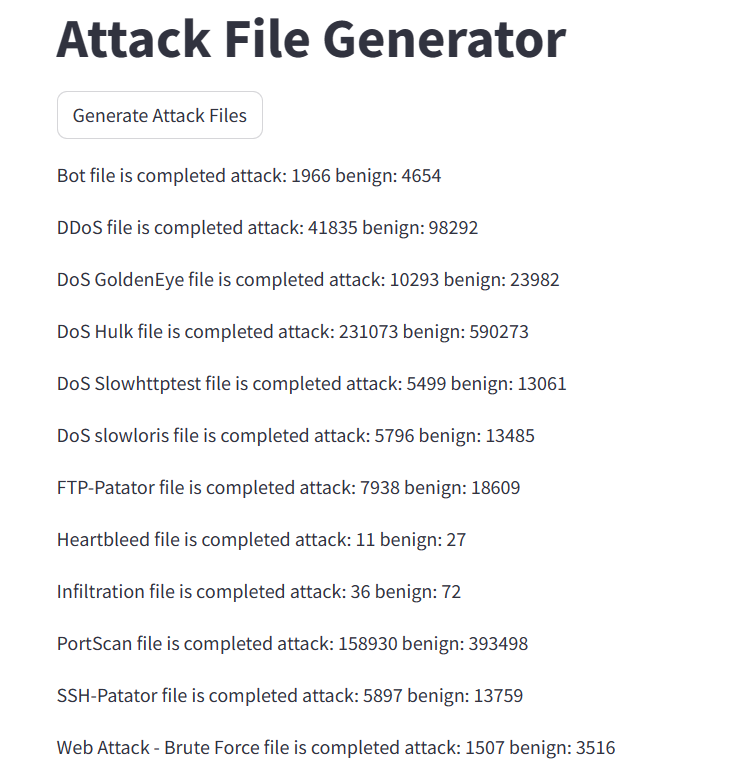
# Result

The study presents a system designed to identify attack data essential for cybersecurity analytics. Users are empowered to specify the types and quantities of attacks to be generated, utilizing the pre-existing dataset named "all\_data.csv." The program orchestrates the creation of attack files for each specified attack type, maintaining a balanced dataset by incorporating a ratio of benign instances. This process involves the systematic reading of a master dataset containing network traffic data, followed by the creation of individual attack files tailored to the specified attack types and quantities of benign instances. Furthermore, the system offers real-time updates on the progress of file generation and merging, ensuring transparency and efficiency throughout the process.

In addition, the application streamlines the analysis by merging web attack files into a single file, thereby facilitating a more cohesive and comprehensive examination of the generated attack data. By providing a user-friendly interface, the code enables easy interaction with the system, enhancing the evaluation and refinement of cybersecurity measures. Overall, the study presents a robust approach to generating diverse attack data, crucial for advancing cybersecurity analytics and fortifying defense mechanisms against evolving cyber threats.

## Analysis

Features to machine learning models and evaluating their F1-score and accuracy. It uses Gaussian Naive Bayes, Quadratic Discriminant Analysis (QDA), and Multi-Layer Perceptron (MLP) classifiers. Features are added one by one, and if a feature improves the F1-score, it's retained; otherwise, it's removed. The process aims to find the most effective feature subset for each classifier. After evaluation, the most efficient feature list for each classifier is printed, along with the corresponding F1-score and accuracy. The code demonstrates an iterative approach to feature selection for optimizing model performance.



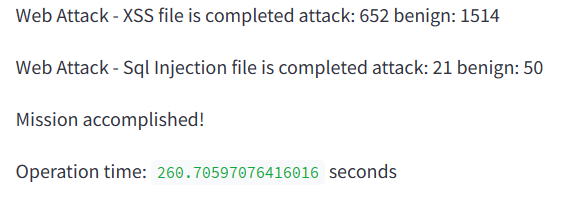


Fig 6. Interface showing attacks and benign

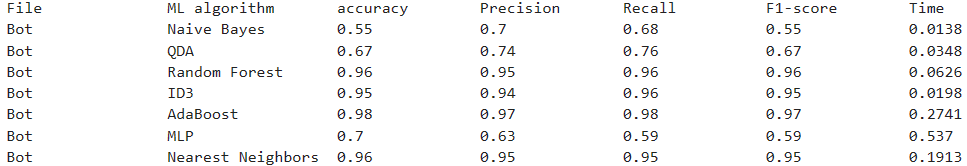


Fig 7. Performance metrics of the model

## Comparison

Wang et al. [10] utilize machine learning techniques for anomaly detection in network security. While the first study proposes the BAE algorithm, leveraging clustering and autoencoder models, achieving F-scores of 96.160, 81.132, and 91.424 on different datasets, the second study employs seven machine learning algorithms on the CICIDS2023 dataset, achieving high accuracy rates ranging from 83% to 97%. Although the BAE algorithm focuses on F-scores, emphasizing precision, recall, and F1-score, the second study prioritizes accuracy, showcasing the effectiveness of diverse machine learning algorithms in detecting network anomalies with high precision. Both studies contribute significantly to enhancing network security through advanced anomaly detection methods.

# Conclusion

In conclusion, the study presents a comprehensive approach to addressing the evolving landscape of internet attacks by advocating for anomaly-based detection methods and leveraging machine learning techniques. Through rigorous evaluation on the CICIDS2023 dataset, the proposed methodology demonstrates high performance rates in detecting network anomalies, with various machine learning algorithms achieving notable success rates. The systematic preprocessing pipeline and model training process lay a solid foundation for robust cybersecurity analytics. Additionally, the implementation of a streamlined application for generating attack data facilitates efficient testing and evaluation of cybersecurity measures. Overall, the study contributes to advancing network security research and enhancing cybersecurity resilience against emerging cyber threats.

# references

1. Engly, A.H., Larsen, A.R., and Meng, W., (2023), “Evaluated anomaly-based intrusion detection”, International Conference on Network and System Security, Vol. 1(5), pp. 277-291.
2. Hussain A., Heidemann, J., and Papadopoulos C.,(2022), “Classifying denial of service attacks at the Conference on Applications, Technologies, Architectures, and Protocols”, Computer communications, Vol. 11(6), pp. 99-110.
3. Vincent P., Larochelle H., and Lajoie I., (2021), “Stacked denoising autoencoders”, Deep networks, Vol. 12(4), pp. 3371-3408.
4. Al-Qatf M., Lasheng Y., and Al-Habib., M.(2024), “Deep learning approach combining sparse autoencoder”, Network intrusion detection Vol. 6(3), pp. 5283-5285.
5. Masci J., Meier U., and Cire D., (2022), “Convolutional autoencoders for hierarchical feature extraction”, Artificial Neural Networks, Vol. 48(5) ,pp.52-59.
6. Park S. and Kim J., (2023), “Network intrusion detection” Stacked denoising autoencoder, Vol. 23(9), pp. 9907-9911.
7. Chen Z., Yeo C.K., and Lee B.S., (2024), “Autoencoder-based network anomaly detection”, Wireless Telecommunications Symposium, Vol. 12(4), pp. 101-578.
8. Shone N., Ngoc T.N., and Phai V.D ., (2021), “Deep learning approach to network intrusion detection”, IEEE Transactions on Emerging Topics in Computational Intelligence, Vol. 22(3), pp. 41-50.
9. Lakhina A., Crovella M., and Diot., C (2023), “Network-wide traffic anomalies”, ACM SIGCOMM Computer Communication Review ,Vol. 34(7), pp. 219-230.
10. Wang, D. and Nie M., (2021), “Machine learning techniques for anomaly detection in network security”, BAE algorithm, clustering and autoencoder, Vol. 42(6), pp. 41-50.