Improving 5g Network Security Using Gan-Based Intrusion Detection with Flask Integration

A. Ruba[1], M.I. Saheedha Sumaya[2], A. Navrin Fathima[3], S. Al Rida[4], S. Al Jameeyathul Sarifa[5]
[1]*Assistant Professor, Department of Artificial Intelligence and Data Science, Mohamed Sathak Engineering College, Kilakarai, Tamil Nadu, India*
[2,3,4,5]*Student, Department of Artificial Intelligence and Data Science, Mohamed Sathak Engineering College, Kilakarai, Tamil Nadu, India*

***Abstract*—The rapid growth of 5G networks and large-scale communication systems has significantly increased the complexity and volume of network traffic, making traditional security mechanisms insufficient to detect modern cyber threats. Intrusion Detection Systems (IDS) play a crucial role in identifying malicious activities and ensuring network security. However, existing IDS solutions often struggle with detecting zero-day attacks and handling imbalanced datasets.This paper proposes a GAN-based Network Intrusion Detection System (NIDS) integrated with a Flask-based web application for real-time monitoring and prediction. The Generative Adversarial Network (GAN) is utilized to generate synthetic attack data, improving model performance and reducing class imbalance issues. Machine learning algorithms such as Random Forest, Naive Bayes, and K-Nearest Neighbour are applied for classification.**
The system is deployed using Flask, enabling user-friendly interaction and real-time intrusion detection. Experimental results demonstrate improved detection accuracy, reduced false alarm rates, and enhanced capability in identifying unknown attacks. The proposed system provides a scalable and efficient solution for securing modern 5G networks.

***Index Terms*—Network Intrusion Detection System (NIDS), GAN, 5G Security, Machine Learning, Flask, Cybersecurity, Anomaly Detection, CICIoT2023, Deep Learning, Data Imbalance**

I. INTRODUCTION

With the advancement of 5G networks and high-speed communication technologies, network security has become a critical concern. The increasing number of connected devices and massive data exchange have led to the emergence of sophisticated cyber threats. Traditional security mechanisms such as firewalls and signature-based intrusion detection systems are no longer sufficient to detect modern attacks, especially zero-day threats.

A Network Intrusion Detection System (NIDS) monitors network traffic to identify suspicious activities and potential security breaches. Machine learning-based IDS solutions have gained popularity due to their ability to learn patterns and detect anomalies in network traffic.

However, one major challenge in IDS is the imbalance in datasets, where normal traffic dominates attack data. This leads to poor detection of rare but critical attacks. To address this issue, this paper proposes a GAN-based intrusion detection model that generates synthetic attack data to improve model training.

Additionally, a Flask-based web application is developed to provide real-time predictions and user interaction, making the system practical for deployment.

II. RELATED WORK

Intrusion Detection Systems (IDS) have been extensively studied over the past decades as a critical component of network security. Early IDS approaches were primarily **signature-based**, relying on predefined attack patterns to detect intrusions. While effective for known threats, these systems failed to identify **zero-day attacks** and required frequent updates, making them less adaptable to dynamic network environments.

To overcome these limitations, researchers introduced anomaly-based detection techniques, which model normal network behavior and identify deviations as potential threats. These approaches improved the detection of unknown attacks but often suffered from high false positive rates, especially in complex and high-dimensional datasets.

With the advancement of Artificial Intelligence, machine learning (ML)-based IDS gained significant attention. Algorithms such as Random Forests (RF), Naive Bayes (NB), Support Vector Machines (SVM), and K-Nearest Neighbors (KNN) have been widely applied for intrusion detection. These models demonstrated strong performance in classifying network traffic into normal and malicious categories. However, their effectiveness heavily depends on feature engineering and balanced datasets, which remain challenging in real-world scenarios.

Recent research has shifted towards deep learning (DL) techniques, including Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN), which can automatically extract complex features from raw network data. These models have shown improved detection accuracy and scalability. Nevertheless, deep learning models require large amounts of labelled data and are computationally expensive.

One of the major challenges in IDS is data imbalance, where normal traffic significantly

outweighs attack traffic. This imbalance leads to poor detection of rare but critical attack types such as U2R and R2L. To address this issue, researchers have explored data augmentation techniques, among which Generative Adversarial Networks (GANs) have emerged as a powerful solution. GANs consist of two neural networks—a generator and a discriminator—that work in opposition to generate realistic synthetic data. By generating additional attack samples, GANs help improve model training and enhance detection performance.

Several recent studies have successfully applied GANs in intrusion detection to improve classification accuracy and reduce false alarm rates. These approaches demonstrate that synthetic data generation can significantly enhance the robustness of IDS models, particularly in handling minority classes.

In parallel, there has been growing interest in integrating IDS systems with web-based frameworks for real-time deployment and accessibility. Frameworks like Flask enable the development of lightweight web applications that allow users to interact with trained models, input network data, and receive predictions in real time. However, many existing systems focus either on model development or deployment, lacking a complete end-to-end solution.

Furthermore, emerging technologies such as 5G networks introduce new security challenges due to their high speed, low latency, and massive device connectivity. Traditional IDS models are not fully equipped to handle the scale and complexity of 5G environments, necessitating more advanced and adaptive approaches.

In this context, the proposed work combines GAN-based data augmentation, machine learning classification, and Flask-based deployment to create a comprehensive intrusion detection system. Unlike previous approaches, this system not only improves detection accuracy through balanced data but also provides a practical, user-friendly interface for real-time monitoring, making it suitable for modern network security applications.

III. SYSTEM DESIGN

The proposed system is designed as a modular and scalable Network Intrusion Detection System (NIDS) that integrates data processing, machine learning, GAN-based augmentation, and a web-based deployment interface. The architecture ensures efficient handling of network traffic data while providing accurate and real-time intrusion detection.
The overall system is divided into the following key modules:

• Data Collection Module
The system utilizes the CICIoT2023 dataset, which contains a wide variety of simulated network traffic records, including both normal and malicious activities. The dataset consists of multiple features such as protocol type, service, duration, source bytes,

and destination bytes.

This module is responsible for:

- Loading and organizing raw network traffic data
- Identifying different types of attacks (DoS, Probe, R2L, U2R)
- Preparing the dataset for preprocessing

The dataset serves as the foundation for training and evaluating the intrusion detection model.

• Data Preprocessing Module

Data preprocessing is a crucial step to ensure the quality and consistency of input data. Raw network data often contains noise, missing values, and inconsistent formats that can negatively impact model performance.

This module performs:

- Data Cleaning: Handling missing or null values
- Feature Selection: Selecting relevant features for model training
- Label Encoding: Converting categorical labels into numerical form
- Feature Scaling: Applying normalization (StandardScaler) to ensure uniform feature distribution
- Train-Test Split: Dividing data into training (80%) and testing (20%) sets

Preprocessing improves model accuracy and ensures stable training by eliminating data inconsistencies.

• System Architecture

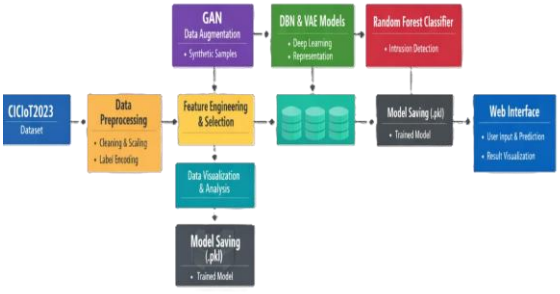

Figure 1: System architecture.

• GAN-Based Data Augmentation Module

One of the major challenges in intrusion detection is class imbalance, where normal traffic samples significantly outnumber attack samples. To address this issue, a Generative Adversarial Network (GAN) is implemented.

The GAN consists of two components:

- Generator: Generates synthetic attack samples that resemble real network traffic
- Discriminator: Distinguishes between real and generated samples

The GAN is trained using:

- Adversarial training approach
- Binary Cross Entropy loss
- Multiple training epochs for convergence

This module helps:

- Increase the number of minority class samples
- Improve model generalization
- Enhance detection of rare attacks

It is important to note that GAN is used only for data augmentation, not for final classification.

• Machine Learning Classification Module

After preprocessing and data balancing, machine learning algorithms are applied to classify

network traffic into normal or attack categories.

The models used include:

- Random Forest Classifier
- Naive Bayes Classifier
- K-Nearest Neighbours (KNN)

These models are trained on the processed dataset and evaluated based on:

- Accuracy
- Precision
- Recall
- F1-Score

Among the models tested, the best-performing model is selected for deployment. The trained model is then saved as a .pkl file for future predictions.

Random Forest Classifier Selection

Among multiple machine learning algorithms (Decision Tree, Naive Bayes, KNN, XGBoost), Random Forest was chosen as the final classifier for

network intrusion detection.

Reason for selection:

- Handles high-dimensional data effectively.
- Reduces overfitting by aggregating multiple decision trees.
- Robust to noisy and imbalanced datasets, which is common in intrusion detection.
- Provides high accuracy and reliable performance across multiple attack types.

Training and Evaluation:

- Dataset split into 80% training and 20% testing
- Performance measured using Accuracy, Precision, Recall, and F1-score
- Achieved improved detection of both known and rare attacks thanks to GAN-augmented data

Outcome:
Random Forest ensured robust, scalable, and real-time-ready classification, making it ideal for deployment in the 5G NIDS system with Flask integration.

• Model Integration Module
The trained machine learning model is integrated into the application for real-time usage. Instead of retraining the model during runtime, the saved model file is loaded directly.

This module:

- Loads the trained model using pickle
- Accepts user input data
- Converts input into required feature format
- Performs prediction (Normal / Attack)

This ensures faster execution and efficient deployment.

• Flask-Based Web Application Module
To make the system user-friendly and interactive, a Flask web application is developed. This module connects the frontend interface with the backend machine learning model.

Key features include:

- User Interface: HTML templates for input and output display
- Jinja2 Templating: Dynamically renders prediction results
- Routing: Handles user requests and responses
- Integration: Connects frontend forms to backend prediction logic

The application allows users to:

- Enter network feature values
- Submit data for analysis
- View prediction results (Attack / Normal)

• System Workflow
The complete system workflow is as follows:

1. User inputs network traffic data through the web interface
2. Input data is sent to the Flask backend
3. Data is pre-processed and formatted
4. Trained model predicts the output
5. Result is displayed on the frontend

Figure 2: System Flow.

• Advantages of the Proposed Design

- Modular and easy to maintain
- Handles data imbalance using GAN
- Supports real-time prediction
- User-friendly interface using Flask
- Scalable for future enhancements

This system design ensures a complete pipeline from data processing to real-time intrusion detection, making it suitable for modern network security applications, especially in 5G environments.

IV. SYSTEM IMPLEMENTATION

The proposed Network Intrusion Detection System (NIDS) is implemented using a combination of machine learning techniques and a web-based deployment framework to ensure both accuracy and usability. The system is designed with a clear separation between the model training phase and the deployment phase for efficient performance.

The backend of the system is developed using Python, where data processing, model training, and prediction logic are handled. The frontend is built using HTML and CSS, integrated through the Flask framework, which enables seamless communication between the user interface and the machine learning model.

Initially, the dataset is preprocessed using libraries

such as Pandas and NumPy. This includes handling missing values, encoding categorical features, and normalizing numerical data using standardization techniques. The processed dataset is then used for training multiple machine learning models, including Random Forest, Naive Bayes, and K-Nearest Neighbors.

To address the issue of class imbalance, a Generative Adversarial Network (GAN) is implemented. The GAN consists of a generator that creates synthetic attack samples and a discriminator that differentiates between real and generated data. Through adversarial training, the GAN produces realistic synthetic data, which is added to the original dataset to improve model performance, especially for minority attack classes.

After training, the best-performing model is selected based on evaluation metrics such as accuracy, precision, recall, and F1-score. The trained model is then saved as a serialized file using the Pickle format (.pkl). This allows the model to be reused without retraining during deployment.

The Flask application serves as the interface for real-time intrusion detection. User inputs are collected through web forms, processed into the required feature format, and passed to the trained model for prediction. The model then classifies the input as either normal traffic or a specific attack type, and the result is displayed on the web interface.

The system also incorporates Jinja2 templating within Flask to dynamically render results on HTML pages. Static resources such as CSS, JavaScript, and images are managed through the static folder, ensuring a responsive and user-friendly interface.

V. RESULTS AND PERFORMANCE EVALUATION

The performance of the proposed GAN-based

Network Intrusion Detection System (NIDS) was evaluated using the CICIoT2023 dataset. The evaluation focuses on measuring the effectiveness of the system in accurately classifying network traffic into normal and various attack categories.

To assess the model performance, standard evaluation metrics were used, including Accuracy, Precision, Recall, and F1-Score. These metrics provide a comprehensive understanding of the model's ability to correctly identify both normal and malicious traffic while minimizing false alarms.

Initially, the machine learning models were trained on the original dataset. However, due to class imbalance, the models showed lower performance in detecting minority attack classes such as R2L and U2R. After applying GAN-based data augmentation, synthetic samples were generated for underrepresented classes, resulting in a more balanced dataset.

The inclusion of GAN-generated data significantly improved the classification performance. The models demonstrated:

- Increased overall accuracy
- Improved detection rate for rare attack types
- Reduction in false positive and false negative rates

Among the implemented models, the Random Forest classifier achieved better performance in terms of accuracy and interpretability, while Naive Bayes and K-Nearest Neighbors also produced satisfactory

results.

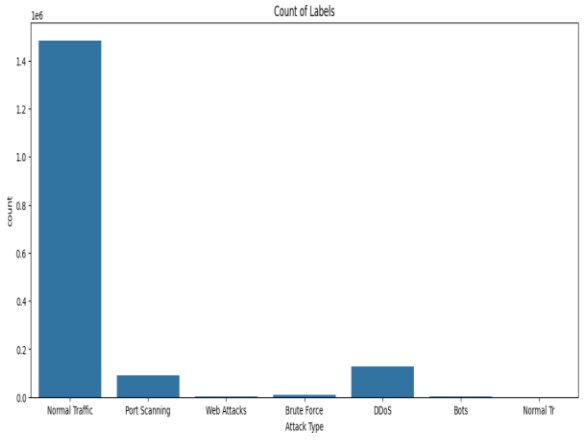

Figure 3: Count of attack type.

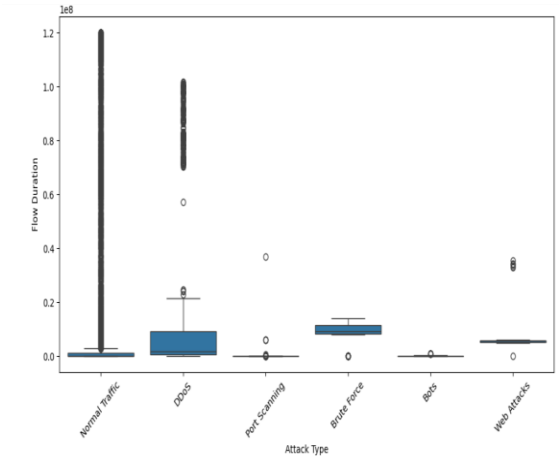

Figure 4: Flow Duration of each attack type.

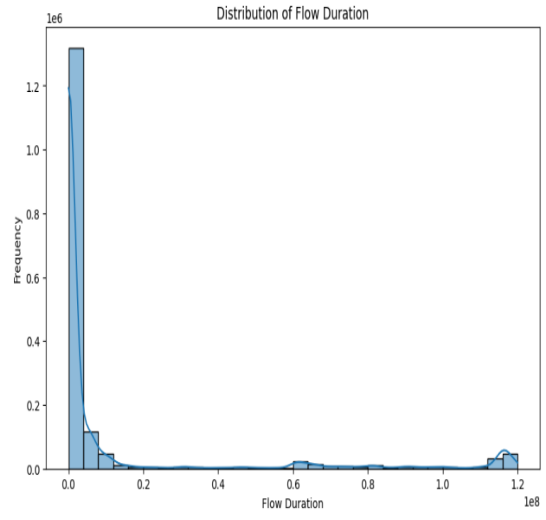

Figure 5: Flow Duration of frequency.

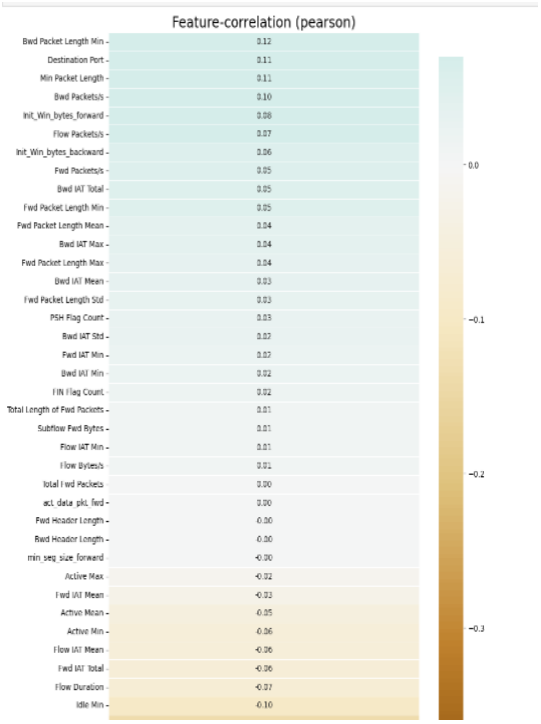

Figure 6: Feature correlation of each attack type.

VI. WEB APPLICATION INTERFACE

A Flask-based web application is developed to provide a user-friendly interface for the intrusion

detection system. The interface allows users to input network traffic features and obtain real-time predictions. The frontend is designed using HTML, CSS, and Bootstrap, while Jinja2 templating is used to dynamically display results. The application ensures seamless interaction between the user and the machine learning model, making the system accessible, efficient, and suitable for practical deployment.

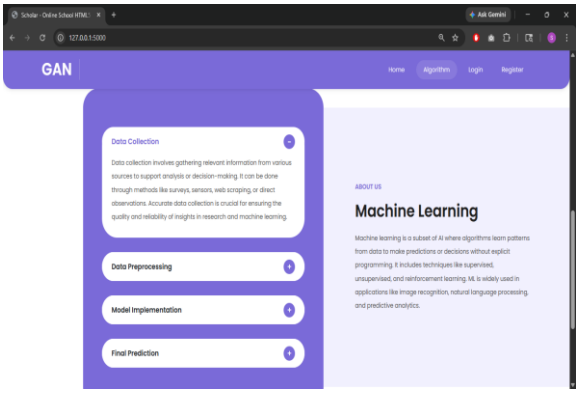

Figure 7: Dashboard page.

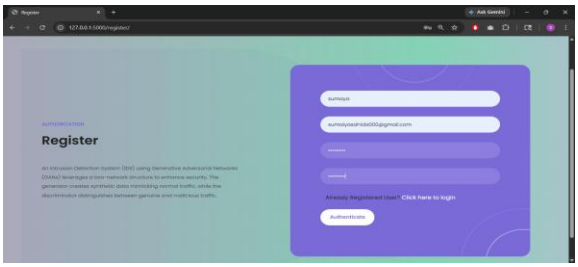

Figure 8: Register page.

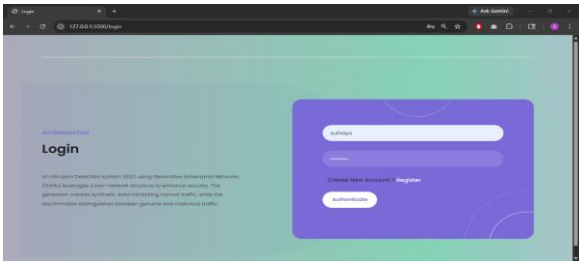

Figure 9: Login page.

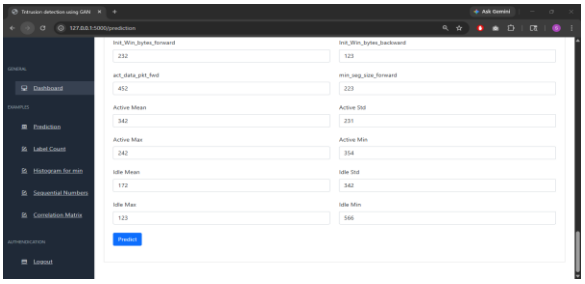

Figure 10: User input page.

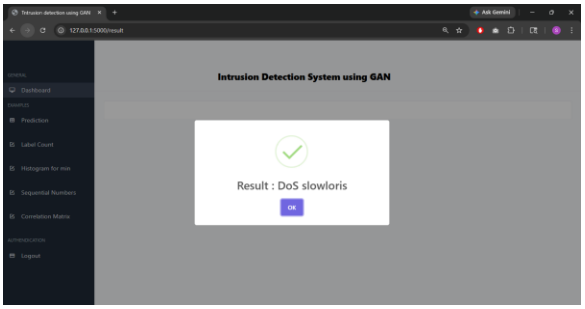

Figure 11: Prediction result is shown.

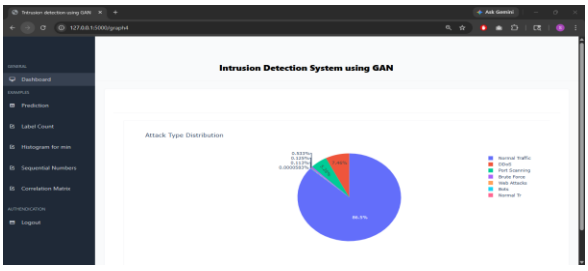

Figure 12: Each attack type comparison.

VII. DISCUSSION

The integration of GAN significantly improves the model's ability to detect rare attacks by balancing the dataset. Compared to traditional methods, the proposed system achieves better accuracy and robustness.

The Flask integration makes the system practical and user-friendly, allowing real-time intrusion detection. However, the system may face limitations when handling extremely large datasets or real-time high-speed traffic.Future improvements can include:

- Deep Learning Models (LSTM, CNN)
- Real-time packet capture
- Cloud deployment

VIII. CONCLUSION

This paper presents a GAN-based Network Intrusion Detection System integrated with Flask for improving 5G network security. The proposed system effectively addresses data imbalance issues and enhances detection accuracy.

The combination of machine learning, GAN, and web deployment provides a scalable and efficient solution for modern cybersecurity challenges. This system can be further extended for real-time large-scale network monitoring.

REFERENCES

[1] Ding, S., Kou, L., & Wu, T. (2022). A GAN-based intrusion detection model for 5G enabled future metaverse. Mobile Networks and Applications, 27(6), 2596-2610.

[2] Moubayed, A. (2024, October). Comparing Boosting-Based and GAN-Based Models for Intrusion Detection in 5G Networks. In 2024 International Symposium on Networks, Computers and Communications (ISNCC) (pp. 1-6). IEEE.

[3] Pandey, C., Tiwari, V., Imoize, A. L., Li, C. T., Lee, C. C., & Roy, D. S. (2023). 5GT-GAN: Enhancing data augmentation for 5G-enabled

mobile edge computing in smart cities. IEEE Access, 11, 120983-120996.

[4] Hussien, M. (2023). Enhancing network security with a deep learning-based intrusion detection system for 5G networks. RS Open Journal on Innovative Communication Technologies, 4(9).

[5] Rabieinejad, E., Yazdinejad, A., Parizi, R. M., & Dehghantanha, A. (2023). Generative adversarial networks for cyber threat hunting in ethereum blockchain. Distributed Ledger Technologies: Research and Practice, 2(2), 1-19.

[6] Sheikh, B. U. H., & Zafar, A. (2024). White-box inference attack: compromising the security of deep learning-based COVID-19 diagnosis systems. International Journal of Information Technology, 16(3), 1475-1483.

[7] King, I. J., Shu, X., Jang, J., Eykholt, K., Lee, T., & Huang, H. H. (2023, October). Edgetorrent: Real-time temporal graph representations for intrusion detection. In Proceedings of the 26th International Symposium on Research in Attacks, Intrusions and Defenses (pp. 77-91).

[8] Jansi Sophia Mary, C., & Mahalakshmi, K. (2024). Modelling of intrusion detection using sea horse optimization with machine learning model on cloud environment. International Journal of Information Technology, 16(3), 1981-1988.

[9] Liang, D., Lu, C. C., Tsai, C. F., & Shih, G. A. (2016). Financial ratios and corporate governance indicators in bankruptcy prediction: A comprehensive study. European journal of operational research, 252(2), 561-572.

[10] Bhatt, R., & Indra, G. (2024). Detecting the undetectable: GAN-based strategies for network intrusion detection. International Journal of Information Technology, 16(8), 5231-5237.

[11] Bhat, K. A., & Sofi, S. A. (2024). A synergistic fusion of shallow and deep generative model to enhance machine learning efficacy and classification performance in data-scarce environments. International Journal of Information Technology, 1-21.

[12] Tiwari, H., Raj, A., Singh, U. K., & Fatima, H. (2024, February). Generative AI for NFTs using GANs. In 2024 11th International Conference on Computing for Sustainable Global Development (INDIACom) (pp. 488-492). IEEE.

[13] Bhat, K. A., & Sofi, S. A. (2024). From shallows to depths: unveiling hybrid synthetic data modeling for enhanced learning with privacy considerations in naturally imbalanced datasets. International Journal of Computers and Applications, 46(12), 1088-1103.

[14] Kumari, S., & Kumar, K. (2023, March). Analysis of Image Enhancement and Deep Learning Techniques. In 2023 10th International Conference on Computing for Sustainable Global Development (INDIACom) (pp. 1365-1370). IEEE.

[15] Gadimov, E., & Birihanu, E. (2025). Real-time suspicious detection framework for financial data streams. International Journal of Information Technology, 1-17.

CITE THIS ARTICLE

A.Ruba, M.I.Saheedha Sumaya, A. Navrin Fathima, S. Al Rida & S. Al Jameeyathul Sarifa, “Improving 5g Network Security Using Gan-Based Intrusion Detection with Flask Integration”.