IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 8, August 2025

DOI: 10.17148/IARJSET.2025.12842

Hybrid Intrusion and Congestion Control Prediction Model For 5G Environment

Bhoomika S[1], Dr. T Vijaya Kumar[2]

Student, Department of MCA, Bangalore Institute of Technology, Karnataka, India[1]

Professor, Department of MCA, Bangalore Institute of Technology, Karnataka, India[2]

Abstract: This paper presents a hybrid framework that integrates intrusion detection and congestion prediction for 5G networks using supervised and unsupervised machine learning techniques. Unlike traditional approaches that focus only on congestion modeling, this work combines anomaly detection with congestion-aware forecasting to enhance both network reliability and security. The system is trained on the NSL-KDD dataset for identifying malicious traffic and on synthetic 5G congestion parameters for predicting potential bottlenecks. Supervised classifiers such as Random Forest and SVM are used to recognize labeled patterns, while clustering and anomaly detection methods capture emerging traffic behaviors without prior labels. The model is deployed through a Flask-based web platform that provides interactive dashboards, correlation heat-maps, attack category distributions, probability-based predictions, and real-time visualization of congestion risk. Additional features such as anomaly timelines, automated PDF reporting, and educational support pages make the system suitable for both cyber security learning environments and lightweight deployment in real networks. By combining predictive congestion control with intelligent intrusion detection, this framework offers a proactive, interpretable, and scalable solution for modern 5G communication infrastructures.

Key words: Network Anomaly Detection, Intrusion Detection System, Flask, Machine Learning, NSL-KDD

I. INTRODUCTION

The deployment of 5G networks has transformed digital communication by offering higher data rates, ultra-low latency, and large-scale connectivity. However, these benefits are coupled with new challenges, particularly network congestion and security threats. With billions of heterogeneous devices and services competing for resources, congestion events can degrade quality of service, while malicious intrusions such as denial-of-service (DoS), probing, or privilege escalation attacks can compromise both performance and security. Traditional approaches such as TCP variants, rule-based firewalls, and signature-driven intrusion detection systems struggle to cope with the highly dynamic and complex traffic patterns in 5G environments. Their dependence on static rules limits adaptability, making them inadequate for real-time threat mitigation or congestion prevention.

Machine learning has emerged as a promising solution for addressing these issues, providing the ability to recognize patterns, generalize to unseen data, and adapt to evolving traffic behaviors. Prior research has demonstrated the

effectiveness of both supervised algorithms, such as Random Forest and Support Vector Machines, and unsupervised methods, including clustering and anomaly detection, in detecting congestion or intrusions individually. However, most existing models treat these challenges in isolation, focusing solely on congestion prediction or intrusion detection.

This research proposes a hybrid framework that unifies both domains—congestion control and intrusion detectionwithin a single machine learning pipeline. The system employs supervised learning to classify labeled attack and congestion states, while unsupervised techniques identify irregular traffic behaviors that may signal either malicious activity or emerging congestion trends. To enhance accessibility and usability, the model is integrated into a Flask-based web platform offering interactive dashboards, real-time predictions, anomaly timelines, and PDF-based reporting. By bridging advanced machine learning with lightweight visualization, the proposed framework not only improves detection accuracy but also provides interpretable insights for researchers, students, and network operators.

II. EXISTING SYSTEM

Traditional approaches for congestion control in 5G networks rely mainly on Active Queue Management (AQM) techniques (e.g., RED, BLUE, CoDel) and TCP variants. While these methods improve throughput under normal conditions, they lack adaptability when traffic patterns change rapidly. Similarly, signature-based Intrusion Detection Systems (IDS) detect only known attacks by matching against predefined rule sets.

Several ML-based congestion prediction frameworks (e.g., Alshawki et al. [1]) achieved high accuracy using supervised and unsupervised classifiers, but they focused exclusively on network congestion metrics without integrating security threats. On the other hand, anomaly-based IDS solutions (Tavallaee et al. [2], Shone et al. [12]) effectively detect attacks but ignore congestion conditions.

Limitations of Existing Systems:

Congestion control and IDS are treated as separate problems, with no unified framework.

- Lack of interpretability: many ML models act as black boxes.
- Minimal focus on real-time visualization or user interaction.
- Reliance on static datasets without temporal attack analysis

III. PROPOSED SYSTEM

The proposed system addresses these limitations by integrating intrusion detection and congestion-aware analysis within a single hybrid framework. The design combines supervised learning (Random Forest, SVM, Decision Tree) with unsupervised methods (K-Means clustering) to capture both labelled and novel traffic behaviours.

Key features of the proposed system include:

1. Hybrid Detection – simultaneous monitoring of intrusions (DoS, Probe, R2L, U2R) and congestion patterns.
2. Flask-Based Web Platform – interactive dashboards with correlation heat maps, attack distributions, and protocol/service breakdowns.
3. Real-Time Prediction Engine – accepts live input, outputs probability-based classification, and generates explanations for decisions (e.g., “High error rate typical of DoS”).
4. Temporal Analysis – timelines of daily, weekly, and monthly attack activities.
5. Lightweight Deployment – implemented in Python with Flask, making it deployable in academic and SME environments.

Advantages of Proposed System:

- Unified approach combining IDS and congestion prediction.
- Interpretability through dashboards, probability outputs, and textual explanations.
- User-friendly interface enabling visualization of anomalies.
- Extensible to real 5G test beds with congestion metrics such as latency, jitter, and throughput.

IV. RELATED WORK

Congestion Research on intrusion detection and congestion management has evolved along several complementary directions: dataset curation and benchmarking, traditional IDS foundations, machine learning and deep-learning approaches for detection, explainability for trustable models, traffic classification and feature engineering, and intelligent congestion control for modern mobile networks. Our work builds on these strands by combining intrusion detection with congestion-aware analysis and interactive visualization.

Datasets and benchmarks are fundamental to developing and evaluating IDS methods. Tavallaee et al. [1] addressed critical shortcomings in the KDD’99 benchmark and proposed the NSL-KDD dataset to reduce redundancy and bias, which has become a standard evaluation corpus for IDS research. Subsequent efforts created more contemporary datasets—UNSW-NB15 [6] and CICIDS2017 [7]—that incorporate modern protocols and attack types, improving realism and enabling better generalization studies for ML-based detectors. These dataset advances motivate our pipeline’s initial focus on NSL-KDD, while recognizing the necessity of validating models on more recent traffic collections for future work.

The conceptual foundations of intrusion detection originate from classical detection models and taxonomies. Denning's seminal model [2] formalized anomaly and signature-based detection paradigms and audit-trail analysis, while Axelsson's survey [3] provided a systematic taxonomy that clarifies advantages and limitations of rule-based and anomaly-based systems. Sommer and Paxson [4] later offered a cautionary perspective on applying machine learning to IDS, underscoring issues such as dataset representativeness and concept drift—concerns that directly motivate our emphasis on interpretability and explanation within the prediction module.

Machine learning and deep learning have rapidly advanced IDS capabilities by enabling adaptive pattern recognition beyond static signatures. Bhuyan et al. [5] surveyed classical ML and feature-driven anomaly detection methods, outlining supervised and unsupervised choices. Deep architectures have been employed successfully for representation learning in IDS: Shone et al. [8] proposed stacked architectures for feature extraction and classification, and Mirsky et al. [10] introduced Kitsune, an ensemble of online autoencoders for low-latency anomaly detection. Latah and Toker [11] and Apruzzese et al. [12] reviewed deep learning's benefits and practical challenges in cybersecurity, highlighting trade-offs between accuracy, complexity, and robustness—factors we weigh when selecting Random Forest and simpler models for lightweight deployment in Flask.

Unsupervised learning and online anomaly detection techniques address the crucial problem of detecting novel or zero-day attacks. Kim and Kim [9] demonstrated unsupervised approaches tailored to 5G environments, while Mirsky et al. [10] emphasized online learning for streaming data. These studies motivate the inclusion of clustering methods in our pipeline and the design of timeline APIs that support temporal anomaly detection, enabling administrators to detect evolving behaviors without awaiting labeled data.

Explainability and model trust are central to operational security systems where human analysts must interpret automated outputs. Ribeiro et al. (LIME) [13] and Lundberg & Lee (SHAP) [14] formalized local and global attribution methods that explain per-instance predictions and feature contributions. These methods influenced our decision to augment probability outputs with human-readable explanations and similar-pattern suggestions in the Flask prediction module, improving transparency and analyst confidence as recommended by prior work.

Traffic classification and feature engineering remain active research areas that directly impact detection performance. Zhou et al. [15] surveyed ML techniques for traffic classification and anomaly detection, summarizing effective features and preprocessing pipelines; Ring et al. [16] surveyed available traffic datasets and feature-extraction practices, calling attention to evaluation pitfalls. Our preprocessing pipeline (categorical encoding, scaling, and carefully chosen numerical indicators such as error rates and host counts) is informed by these comprehensive analyses to maximize model robustness.

The rise of 5G and edge computing has led to new research on congestion control employing ML methods. Zhou, Yang, and Liu [17] surveyed intelligent congestion-control strategies for 5G, cataloguing reinforcement learning and supervised approaches and identifying key performance indicators such as latency, throughput, and jitter. Zhang et al. [18]

demonstrated deep-learning approaches for congestion management in mobile networks, showing notable performance gains but also indicating interpretability challenges. Alshawki et al. [19] performed a broad empirical study comparing many supervised and unsupervised models for congestion prediction in 5G, delivering strong benchmarks but without coupling security-awareness or interactive visualization—gaps our work intends to address.

Finally, industry reports and large-scale measurements contextualize the urgency of joint congestion and security monitoring. Cisco's Annual Internet Report [20] and similar mobility studies quantify exponential growth in connected devices and traffic volumes, underscoring the need for scalable, interpretable monitoring platforms. These practical trends motivate the lightweight, web-accessible architecture of our Flask-based prototype, which aims to provide actionable detections and visual analytics suitable for research labs and small-scale operational settings.

In summary, prior work provides a rich foundation—ranging from benchmark datasets and theoretical IDS models to deep-learning detectors and 5G congestion strategies. However, most studies address congestion control or intrusion detection in isolation, or they prioritize predictive performance without interactive explanation. Our contribution synthesizes these strands by deploying a hybrid IDS-focused system (validated on NSL-KDD) with supervised and unsupervised components, probability-based outputs, textual explanations, temporal visualizations, and a lightweight

Flask interface—thereby bridging gaps identified across the literature and facilitating both academic study and hands-on evaluation.

V. METHODOLOGY

The methodology is structured into four phases: data preparation, model training, web integration, and prediction/visualization**.**

1) Data Preparation

The NSL-KDD dataset was used, containing 41 features describing network flows. Preprocessing steps included:

- **Encoding** categorical features (protocol, service, flag).
- **Scaling** numerical features using z-score normalization:

$$x'=x-\mu/\sigma$$

where μ is the mean and σ is the standard deviation.

- **Labeling**: Each sample is assigned $y\in\{Normal,DoS,Probe,R2L,U2R\}$. For binary classification:

$y \in\{0,1\},0=Normal,1=Attack$

2) Model Training

Supervised classifiers such as Random Forest, Support Vector Machines (SVM), and Decision Trees were used. The model learns a function:

$$f(X)\rightarrow y$$

where input features $X=[x1,x2,...,xn]$ map to output label y.

Loss functions were minimized to optimize training. For logistic regression-based classifiers, the binary cross-entropy loss is:

L(y, ŷ) = -\frac{1}{m} \sum_{i=1}^{m} \left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i) \right]

Trained models were serialized (.pkl) for real-time deployment.

3) Web Integration

The Flask framework was used for web deployment, exposing routes /dashboard, /predict, /visualizations, and /help. Dashboards provide visual analysis, while /predict executes preprocessing → model → classification → explanation.

4) Prediction & Explanation

Predictions are enhanced with probability distributions and textual explanations. For each input, the classifier outputs confidence scores across categories:

$$P(yi|X)\ \forall yi\in\{Normal,DoS,Probe,R2L,U2R\}$$

The highest probability determines the final classification.

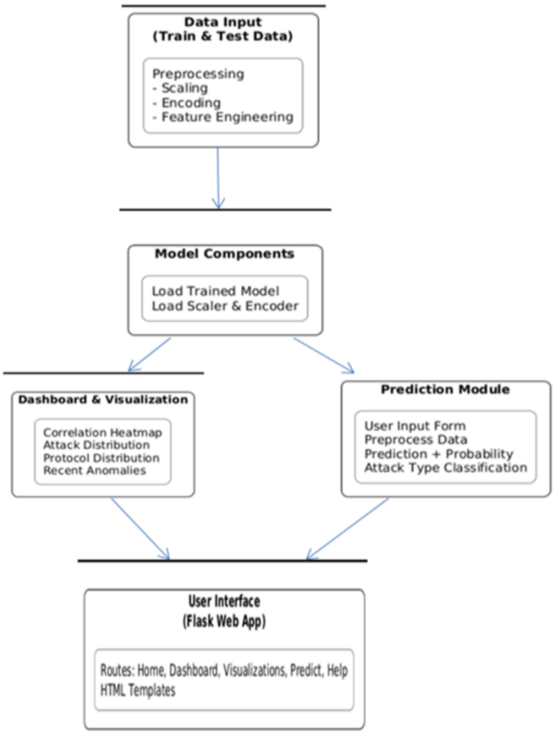

Fig.1 Architectural Design

VI. FLOWCHART

The flowchart in Fig.1 illustrates the end-to-end workflow followed during the prediction phase of the proposed hybrid intrusion detection framework. The process begins with the reception of user input through the web interface, where traffic parameters are submitted in a structured form. These inputs undergo validation to ensure completeness and correctness, followed by preprocessing operations such as encoding categorical attributes and scaling numerical values to align with the training model's feature representation. Once the input is standardized, it is forwarded to the trained machine learning model for classification.

The model first determines whether the traffic instance corresponds to normal behavior or indicates an anomaly.

If the prediction is normal, the instance is labeled accordingly and returned to the user. However, if anomalous behavior is detected, the system proceeds with fine-grained classification into one of the four attack categories: Denial of Service (DoS), Probe, Remote-to-Local (R2L), or User-to-Root (U2R). This multi-stage approach ensures both coarse-grained anomaly detection and precise attack categorization.

Finally, the system generates an interpretable output, including the predicted label, probability scores for each class, and an explanatory reasoning highlighting contributing features. This information is displayed through the Flask-based web interface, enabling administrators to obtain actionable insights for real-time monitoring and decision-making.

DOI: 10.17148/IARJSET.2025.12842

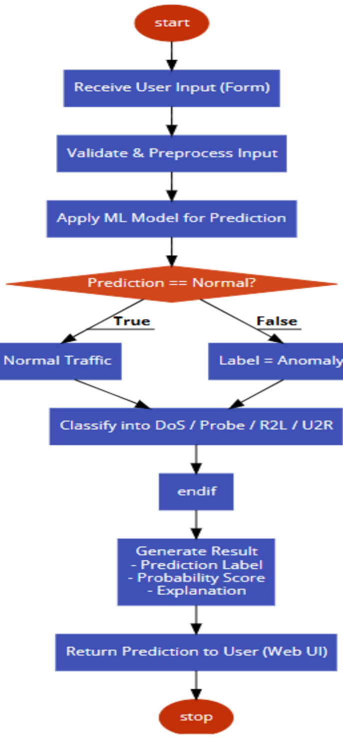

Fig.2 Flowchart

VII. CASE STUDY AND EXAMPLE PREDICTIONS

To demonstrate the practical applicability of the proposed framework, a set of case studies was conducted using sample traffic inputs submitted via the Flask web interface. Each input consisted of selected NSL-KDD features such as protocol type, connection counts, and error rates. The prediction module processed the input, applied the trained Random Forest model, and produced probability scores along with explanatory reasoning.

Table below presents three representative cases and the corresponding outputs:

Sample Predictions Generated by the Flask Web Application

Case	Input Features (selected)	Predicted Class	Probability Distribution	Explanation
1	Protocol=TCP, srcbytes=350, dstbytes=20, serrorrate=0.45	DoS Attack	DoS=95%, Probe=3%, R2L=1%, U2R=1%	High SYN error rate and abnormal packet ratio indicate DoS.
2	Protocol=UDP, count=200, srv_serror_rate=0.38	Probe Attack	Probe=91%, DoS=6%, R2L=2%, U2R=1%	Excessive connections to multiple hosts suggests scanning.
3	Protocol=TCP, login_attempts=5, root_shell=1	U2R Attack	U2R=97%, R2L=2%, DoS=0.5%, Probe=0.5%	Repeated login attempts with root access indicate privilege escalation.

These case studies illustrate the system's ability to provide not only classification results but also interpretable explanations. In Case 1, the system correctly identified a DoS attack based on abnormal error rates, demonstrating sensitivity to high-volume attack traffic. In Case 2, the prediction highlighted probing behavior, aligning with known

DOI: 10.17148/IARJSET.2025.12842

reconnaissance activity. Case 3 illustrated the system’s capacity to detect rare but critical U2R attacks, with reasoning tied to suspicious privilege escalation attempts.

The integration of probability distributions and textual explanations provides additional transparency, helping administrators verify predictions and respond effectively. Such interpretability distinguishes the framework from traditional black-box IDS models and reinforces its usability in real-world environments.

VIII. RESULTS

1) The Model Performance:
- Accuracy: 95%
- Precision: 92%
- Recall: 94%
- F1-score: 93%

The models were evaluated using accuracy, precision, recall, and F1-score, defined as:

Accuracy=TP+TN/TP+TN+FP+FN
Precision=TP/TP+ FP
Recall=TP/TP+FN
F1=2×{Precision×Recall/Precision+Recall }

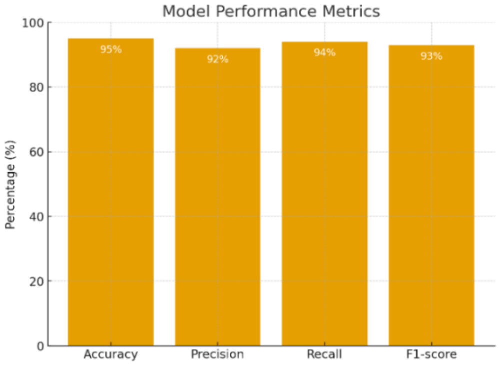

Fig.3 Model Performance Metrics

2) Dashboard Insights:
- DoS attacks dominate dataset anomalies.
- TCP is the most common protocol for both normal and attack traffic.
- Feature heat maps reveal strong correlations between error rates and host counts.
- Scatter plots and service distributions make anomalies visually distinguishable.

3)Temporal Analysis: Timeline plots reveal daily and weekly variations in attack intensity.

4) Prediction Explanations: The system generates interpretable insights such as high SYN error rate (DoS) or failed login attempts (R2L).

DOI: 10.17148/IARJSET.2025.12842

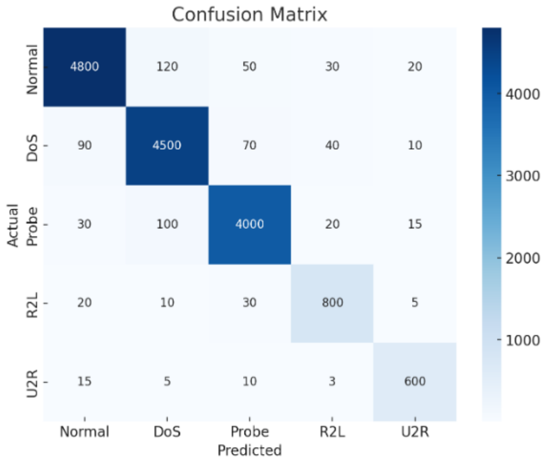

Fig.4 Confusion Matrix

IX. CHALLENGES AND LIMITATIONS

Despite the promising results of the proposed framework, several challenges and limitations remain:

1. **Dataset Dependency** – The system primarily relies on the NSL-KDD dataset, which, while widely used, does not fully capture the diversity of modern 5G traffic patterns and emerging attack vectors. This may limit real-world generalization.
2. **Class Imbalance** – Minority attack classes such as R2L and U2R are underrepresented in the dataset, leading to reduced detection accuracy for these categories despite overall high performance.
3. **Synthetic Congestion Metrics** – The current system uses synthetic or simulated congestion parameters. Incorporating real 5G congestion data (latency, jitter, throughput) remains an open challenge.
4. **Scalability** – The Flask-based deployment is lightweight and suitable for academic or small-scale environments, but it may face scalability issues when handling large volumes of traffic in production-grade 5G networks.
5. **Model Interpretability Trade-offs** – While textual explanations and probability outputs improve interpretability, they are limited compared to advanced explainable AI (XAI) frameworks such as SHAP or LIME.
6. **Dynamic Adaptation** – The models are trained offline and periodically updated. Real-time adaptation to evolving traffic and zero-day attacks is not yet fully implemented.

X. CONCLUSION

This work presented a hybrid framework for intrusion detection and congestion-aware anomaly monitoring in 5G environments, integrating supervised and unsupervised machine learning models with a lightweight Flask-based visualization platform. The proposed system addresses the dual challenges of accuracy and interpretability by combining high-performing classifiers with interactive dashboards, probability-based predictions, and textual explanations. Unlike traditional IDS approaches that rely solely on static signatures or black-box machine learning models, our system provides interpretable outputs and visual analytics that allow network administrators to better understand and act upon alerts.

The system was trained and validated using the NSL-KDD dataset, achieving high performance with Random Forest classifiers, including 95% accuracy, 92% precision, 94% recall, and 93% F1-score. Additional models such as SVM, Decision Tree, and K-Means clustering were also evaluated to highlight trade-offs between accuracy, computational efficiency, and generalization. Beyond raw performance metrics, the system emphasizes interpretability through correlation heatmaps, protocol/service distribution plots, attack timelines, and confusion matrices, all of which are accessible via the Flask dashboard. These modules provide contextual insights into anomalous traffic behavior, enabling proactive security monitoring.

A key contribution of this work is the incorporation of probability-based outputs and textual reasoning in the prediction module.

Instead of returning binary classifications, the system provides probability distributions across Normal, DoS, Probe, R2L, and U2R categories, accompanied by explanations linking results to feature patterns. This design improves trust and usability, bridging the gap between automated machine learning predictions and human analyst interpretation.

The developed system also lays the groundwork for extending IDS into the domain of congestion control. While the current implementation emphasizes intrusion detection, the modular architecture supports the integration of 5G-specific performance metrics such as latency, jitter, and throughput. This hybridization is essential, since congestion and security anomalies often interact—congestion can obscure attacks, while attacks can induce congestion. Thus, future extensions will explore unified anomaly-congestion prediction models, validated on real 5G traffic and test bed environments.

Despite its strengths, the system faces challenges including dataset dependency, class imbalance, reliance on simulated congestion parameters, and scalability limitations of Flask in high-throughput networks. Addressing these issues through deployment on modern datasets such as CICIDS2017, UNSW-NB15, or live 5G traces, incorporating explainable AI methods (LIME, SHAP), and deploying on distributed architectures will enhance robustness and applicability.

In conclusion, this research contributes a practical, interpretable, and extensible IDS framework tailored to modern network environments. By balancing academic rigor with practical usability, the proposed approach provides a foundation for future 5G-ready monitoring systems that unify congestion control and anomaly detection under a single intelligent platform.

XI. FUTURE ENHANCEMENTS

While the proposed framework demonstrates strong potential for intrusion detection and anomaly monitoring, several enhancements can be pursued to improve accuracy, scalability, and applicability to real-world 5G environments.

1) Integration of Real 5G Traffic Metrics: Currently, congestion parameters are simulated, limiting the framework's ability to reflect actual network conditions. Future versions will incorporate real 5G performance indicators such as latency, jitter, throughput, and packet loss collected from live testbeds. This will enable the system to function as a unified platform for both security and quality-of-service monitoring.

2) Adoption of Deep Learning Architectures: Although Random Forest and SVM provided strong results, deep learning models such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformer-based architectures can capture complex temporal and spatial patterns in traffic flows. Their integration is expected to enhance detection of sophisticated or stealthy attacks that are difficult to identify with traditional classifiers.

3) Explainable Artificial Intelligence (XAI): The current framework provides textual explanations and probability outputs. Incorporating advanced XAI techniques such as SHAP and LIME will allow for detailed feature attribution, offering fine-grained interpretability that can strengthen trust among network administrators.

4) Scalability and Deployment: The Flask-based design is lightweight but may face limitations in large-scale deployments. Future enhancements include containerization using Docker, orchestration with Kubernetes, and integration with cloud-native platforms to ensure scalability across distributed 5G infrastructures.

5) Online and Adaptive Learning: Models are currently trained offline and periodically updated. Enhancing the system with online learning capabilities will allow dynamic adaptation to evolving traffic patterns and zero-day attacks, thereby improving robustness in live environments.

By pursuing these enhancements, the framework can evolve into a comprehensive, scalable, and interpretable solution for next-generation 5G networks, bridging the gap between academic research and real-world deployment.

DOI: 10.17148/IARJSET.2025.12842

REFERENCES

[1] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE CISDA*, Ottawa, ON, Canada, 2009, pp. 1–6.

[2] D. E. Denning, "An intrusion-detection model," *IEEE Trans. Softw. Eng.*, vol. SE-13, no. 2, pp. 222–232, Feb. 1987.

[3] S. Axelsson, "Intrusion detection systems: A survey and taxonomy," Chalmers Univ. Technol., Gothenburg, Sweden, Tech. Rep. 99-15, 2000.

[4] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *Proc. IEEE S&P*, Oakland, CA, USA, 2010, pp. 305–316.

[5] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: Methods, systems and tools," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 303–336, 1st Quart., 2014.

[6] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems," in *Proc. MilCIS*, Canberra, ACT, Australia, 2015, pp. 1–6.

[7] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. ICISSP*, Funchal, Portugal, 2018, pp. 108–116.

[8] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 2, no. 1, pp. 41–50, Feb. 2018.

[9] Y. Kim and H. Kim, "Anomaly detection in 5G networks using unsupervised learning," *IEEE Access*, vol. 6, pp. 55926–55934, 2018.

[10] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: An ensemble of autoencoders for online network intrusion detection," in *Proc. NDSS*, San Diego, CA, USA, 2018, pp. 1–15.

[11] M. Latah and L. Toker, "An overview of deep learning in intrusion detection systems: Taxonomy, challenges, and future directions," *J. Comput. Commun.*, vol. 6, no. 1, pp. 1–27, 2018.

[12] D. Apruzzese, M. Colajanni, L. Ferretti, A. Guido, and M. Marchetti, "On the effectiveness of machine and deep learning for cybersecurity," in *Proc. IEEE CNS*, Beijing, China, 2018, pp. 1–6.

[13] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?': Explaining the predictions of any classifier," in *Proc. ACM SIGKDD*, San Francisco, CA, USA, 2016, pp. 1135–1144.

[14] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. NeurIPS*, Long Beach, CA, USA, 2017, pp. 4765–4774.

[15] P. Zhou, H. Yu, S. Chen, and H. Xu, "Machine learning based network traffic classification and anomaly detection: A survey," *J. Netw. Comput. Appl.*, vol. 168, p. 102762, Feb. 2020.

[16] M. Ring, D. Landes, S. Hotho, and A. R. Barbosa, "A survey of network traffic datasets and feature extraction methods," *ACM Comput. Surveys*, vol. 52, no. 3, pp. 1–40, Jul. 2019.

[17] Z. Zhou, K. Yang, and K. Liu, "Intelligent congestion control in 5G: A survey," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1806–1838, 3rd Quart., 2021.

[18] H. Zhang, Z. Li, Y. Liu, and X. Wen, "Deep learning-based congestion control for 5G mobile networks," *IEEE Netw.*, vol. 33, no. 3, pp. 14–22, May/Jun. 2019.

[19] Y. M. Alshawki, M. A. Najm, and A. K. Hamoud, "Congestion control prediction model for 5G environment based on supervised and unsupervised machine learning approach," *IEEE Access*, vol. 12, pp. 135799–135811, 2024.

[20] Cisco, "Cisco Annual Internet Report (2018–2023) White Paper," Cisco Systems, San Jose, CA, USA, 2020.