StreamGuard: A Real-Time IDS Pipeline Leveraging Apache Kafka

Karthikeyan B
Student(UG),Dept of Computer Science and Engineering (Artificial Intelligence and Machine Learning)
Vel Tech High Tech Dr. Rangarajan Dr. Sakunthala Engineering College
Chennai,India
vishalkarthikeyan44@gmail.com

K Jai Akash
Student(UG),Dept of Computer Science and Engineering (Artificial Intelligence and Machine Learning)
Vel Tech High Tech Dr. Rangarajan Dr. Sakunthala Engineering College
Chennai,India
peligroakash345@gmail.com

Pavin B
Student(UG),Dept of Computer Science and Engineering (Artificial Intelligence and Machine Learning)
Vel Tech High Tech Dr. Rangarajan Dr. Sakunthala Engineering College
Chennai,India
pavinb2782004@gmail.com

Dr.Queen Mary Vidya M
Asst. Prof.,Dept of Computer Science and Engineering (Artificial Intelligence and Machine Learning)
Vel Tech High Tech Dr. Rangarajan Dr. Sakunthala Engineering College
Chennai,India
queenmaryvidya@velhightech.com

Abstract— StreamGuard is a real-time intrusion detection framework developed to identify malicious network activities in high-speed streaming environments. The proposed system combines machine learning techniques with distributed stream processing to support continuous and low-latency cyber threat detection. Apache Kafka is used as the streaming backbone to capture and transport network traffic data in real time, ensuring reliable processing under high-throughput conditions. To improve computational efficiency, Principal Component Analysis (PCA) is applied to reduce feature dimensionality prior to classification. The framework is evaluated using the CSE–CIC–IDS2018 dataset and includes a comparative study of multiple machine learning models such as Random Forest, XGBoost, Gaussian Naïve Bayes, and Stochastic Gradient Descent (SGD). Based on experimental results and real-time processing requirements, the SGD classifier is selected as the final detection model due to its strong generalization capability and fast inference performance. StreamGuard also integrates InfluxDB for storing streaming detection results and Grafana dashboards to provide real-time visualization of network activity and alerts for effective monitoring.

Keywords–- Intrusion Detection System, Apache Kafka, Stochastic Gradient Descent, PCA, Real-Time Monitoring, Cybersecurity

INTRODUCTION

Cybersecurity has emerged as a fundamental challenge for governments, enterprises, and individual users as modern digital infrastructures continue to expand in scale and complexity. The rapid proliferation of cloud computing platforms, Internet of Things (IoT) ecosystems, and large-scale enterprise networks has significantly increased the volume, velocity, and diversity of network traffic.

While these advancements enable greater connectivity and operational efficiency, they simultaneously enlarge the attack surface, making contemporary networks more vulnerable to persistent and sophisticated cyber threats. Modern cyberattacks are no longer sporadic or static in nature; instead, they are continuous, adaptive, and increasingly automated. Adversaries employ advanced techniques such as polymorphic malware, distributed denial-of-service attacks, and intelligent evasion strategies that dynamically alter attack behavior to bypass conventional defense mechanisms. Traditional security solutions—including firewalls, antivirus software, and signature-based intrusion detection systems—primarily rely on predefined rules or known attack signatures. Although effective against previously observed threats, such approaches struggle to detect zero-day attacks and novel intrusion patterns, particularly in high-speed and evolving network environments. To overcome these limitations, recent research has increasingly focused on machine learning–based intrusion detection systems that learn patterns from network traffic data and identify anomalous behavior automatically. While these approaches demonstrate improved detection capabilities compared to rule-based systems, many existing solutions are designed for offline or batch-oriented processing. As a result, they are poorly suited for real-time deployment, where rapid detection and response are critical for minimizing the impact of cyberattacks. Additionally, complex models with high computational overhead often introduce latency, limiting their practicality in streaming scenarios.These challenges highlight the need for intrusion detection frameworks that are not only accurate but also scalable, lightweight, and capable of operating continuously on streaming data.

In this context, StreamGuard is proposed as a real-time intrusion detection framework that integrates distributed stream processing with efficient machine learning–based traffic classification The system leverages Apache Kafka as a high-throughput streaming backbone to enable continuous ingestion and processing of network traffic, supporting low-latency analysis under high-load conditions. To ensure computational efficiency in real-time inference, Principal Component Analysis (PCA) is employed to reduce feature dimensionality while retaining essential traffic characteristics. Unlike systems that rely on computationally intensive ensemble or deep learning models, StreamGuard adopts a Stochastic Gradient Descent (SGD)–based classifier as the core detection mechanism. The model is trained and evaluated using the CSE–CIC–IDS2018 dataset and is selected through a comparative analysis against multiple machine learning algorithms, including Random Forest, XGBoost, and Gaussian Naive Bayes. Experimental evaluation demonstrates that the SGD classifier provides a favorable balance between detection performance and execution efficiency, making it particularly suitable for real-time streaming intrusion detection.

In addition to automated detection, the system integrates InfluxDB as a time-series database for storing prediction results and network metrics generated during real-time analysis. These data are visualized through Grafana dashboards, which provide interactive monitoring of network activity, model predictions, and system performance metrics. This visualization layer enables security administrators to observe traffic patterns, identify abnormal behavior, and respond quickly to potential threats. By combining real-time data streaming, lightweight machine learning, time-series storage, and interactive visualization, StreamGuard supports a proactive and scalable intrusion detection approach suitable for modern distributed network environments.

LITERATURE SURVEY

Recent research in cybersecurity has increasingly emphasized the use of machine learning techniques to enhance the accuracy, adaptability, and responsiveness of intrusion detection systems. Traditional signature-based and rule-driven security mechanisms struggle to identify evolving attack patterns and zero-day threats, motivating the shift toward data-driven detection models capable of learning complex and dynamic network behaviors. Machine learning–based IDS approaches enable automated analysis of large-scale network traffic and improve detection performance compared to static security solutions.

Despite these advancements, many existing intrusion detection approaches rely on offline analysis or focus only on isolated stages of the detection pipeline, limiting their effectiveness in real-time deployments. There remains a need for an end-to-end intrusion detection framework that seamlessly integrates real-time data streaming, efficient feature processing, lightweight machine learning classification, and intuitive visualization. The proposed StreamGuard system addresses these challenges by combining Apache Kafka–based streaming with PCA-assisted Stochastic Gradient Descent classification and real-time monitoring, offering a scalable, low-latency, and practical solution for modern network security environments..

Table I
Literature Survey

S.NO	TITLE OF THE PAPER	ALGORITHM USED	DATASET	INFERENCE	YEAR OF PUBLICATION
1	Stochastic Gradient Descent Classifier-Based Lightweigh t Intrusion Detection System	Stochastic Gradient Descent (SGD) Classifier	Network Traffic Dataset (CSE–CIC–IDS / Benchmark IDS Data)	Highlights the effectivene ss of SGD-based classifiers for efficient real-time intrusion detection.	2024
2.	AI-Driven Intrusion Detection System Using Kafka Stream Processing and Federated Learning	Federated Learning + Deep Neural Network	UNS WNB 15, NSLK DD	Proposes a federated deep learning model integrated with Kafka for decentralised intrusion detection, ensuring data privacy while maintaining high detection accuracy across distributed nodes	2025
3	CND-IDS: Continua Novelty Detection for Intrusion Detection Systems	Continual Learning + Novelty Detection (Unsupervised)	Various Realistic Intrusion Streams	Introduces a continual learning framework addressing concept drift and unseen attacks in streaming intrusion detection response.	2025
4	Toward Secure SDN Infrastructure in Smart Cities: Kafka-Enabled Machine Learning Framework for Anomaly Detection	Random Forest, XGBoost, Linear Regression (Ensemble) + Kafka Streaming	InSDN Intrusion Detection Dataset	Proposes a Kafka-based ML framework for SDN anomaly detection, achieving reduced detection latency and high scalability for edge environments	2025
5	AE-Integrated: Real-time Network Intrusion Detection with Apache Kafka and Autoencoder	Autoencoder (Deep Learning Anomaly Detection)-	CICI DS20 17	A streaming deep learning approach using Kafka for real-time anomaly detection. The study demonstrat ed high accuracy for identifying intrusion attempts.	2024

6	Machine Learning-based Network Intrusion Detection for Big and Imbalanced Data using Oversampling, Stacking Feature Embedding, and Feature Extraction	Machine Learning (Oversampling + Stacking + Feature Embedding)	MultipleIDS datasets (imbalanced and large-scale	Demonstrates improved performance in detecting intrusions on large, imbalanced datasets using an advanced ML pipeline.	2024

METHODS AND MATERIALS

A. Dataset

Access to real-world network traffic data for cybersecurity research is often constrained by privacy, legal, and ethical considerations. To address these challenges while maintaining realism and reproducibility, the proposed StreamGuard framework utilizes the CSE–CIC–IDS2018 dataset, a widely recognized benchmark developed by the Canadian Institute for Cybersecurity (CIC) in collaboration with the Communications Security Establishment (CSE). The dataset emulates realistic enterprise network behavior and is extensively used for evaluating intrusion detection systems. The CSE–CIC–IDS2018 dataset contains labeled network flow records representing both benign and malicious activities. In this work, traffic corresponding to Denial-of-Service (DoS) attack scenarios and benign network communication collected over multiple days (Tuesday to Friday) was utilized. These attack types represent high-impact and frequently observed threats in real-world networks, making them suitable for evaluating real-time intrusion detection performance.

Dataset Preprocessing and Selection-

The selected dataset consists of approximately 3.95 million network flow records with 100 features. Preprocessing steps included the removal of non-informative attributes such as timestamp fields, handling missing and infinite values, and retaining only numerical features suitable for machine learning models. To ensure robustness and consistency, all undefined and infinite values were replaced during preprocessing. The processed dataset was divided into training and testing sets using an 80:20 split, resulting in approximately 3.16 million samples for training and 0.79 million samples for testing. Feature scaling was applied prior to dimensionality reduction, and Principal Component Analysis (PCA) was employed to retain 95% of the original variance, reducing computational overhead while preserving essential traffic characteristics.

Feature Engineering and Dimensionality Reduction-

Each network flow is represented as a structured feature vector containing statistical attributes such as packet counts, byte rates, flow duration, and header information. Since high-dimensional feature spaces can introduce redundancy and increase computational overhead, Principal Component Analysis (PCA) was applied to reduce dimensionality while retaining 95% of the variance in the data. This step enhances classification efficiency and improves real-time performance without compromising detection accuracy.

Dataset Storage and Integration-

The processed dataset is stored in a structured comma-separated values (CSV) format, allowing seamless integration with machine learning workflows and Kafka-based streaming components. The data is partitioned into training and testing sets using a stratified split to preserve class distribution across both sets. This organisation supports repeated model training, evaluation, and real-time traffic simulation within the StreamGuard architecture.

Significance of the Dataset-

The adopted dataset strategy provides the following advantages:

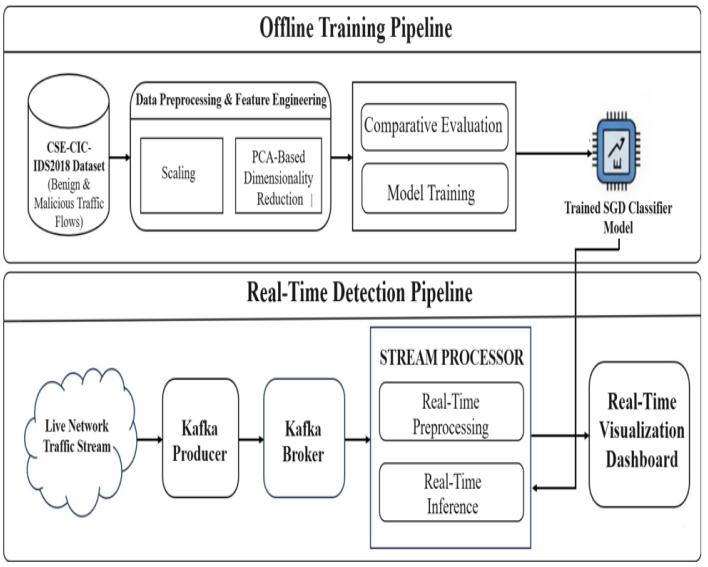
- Ensures realistic representation of enterprise network traffic
- Eliminates dependence on sensitive real-world traffic captures
- Supports repeatable and scalable experimentation
- Enables efficient real-time streaming and classification
- Provides a reliable foundation for PCA-assisted SGD-based intrusion detection

Table II
Description Of The Attributes

Attribute Name	Data Type	Description
Timestamp	Date & Time	Records the exact time at which the network traffic flow was captured during real-time monitoring
Source Port	Integer	Port number from which the attacker started the connection
Destination Port	Integer	Target service port of the destination server receiving the network request
Packet Rate	Integer	Number of packet or requests generated per unit time, indicating attack intensity

Protocol	Categorical	Network protocol used for communication.
Attack Type	Categorical	Class label representing the nature of the cyberattack
Attack Label	Integer	Numerically encoded representation of the attack type for machine learning model.
Defense Action	Categorical	Mitigation strategy triggered by the system.

Figure I
Architecture Diagram



B. Methodology

System Architecture and Functional Components-

The proposed StreamGuard framework is designed as a real-time intrusion detection system that focuses on continuous monitoring, efficient traffic analysis, and timely identification of potential cyber threats without interrupting normal network operations. Unlike conventional security approaches that depend on static rules or post-incident analysis, StreamGuard follows a streaming-oriented architecture that enables proactive detection of malicious network activities as they occur in real time.

The first layer of the system in Figure I handles data acquisition and traffic streaming. Network flow records derived from the CSE–CIC–IDS2018 dataset are used to represent realistic enterprise network behavior, including both benign and malicious traffic patterns. To simulate live network conditions, the dataset is streamed using Apache Kafka, which converts static flow records into continuous data streams that resemble real-time network traffic and allow the system to process events in a streaming environment.

At the core of the architecture lies the stream processing component, which continuously consumes network flows from the Kafka broker. During online operation, incoming traffic is transformed using the same feature representation learned during the offline training phase. In particular, feature scaling and Principal Component Analysis (PCA) transformation are applied to ensure consistency between training and real-time inference. PCA reduces the dimensionality of the feature space while preserving the most informative traffic characteristics, thereby improving computational efficiency and enabling low-latency detection in streaming environments. The transformed feature vectors are then passed to the machine learning layer, where a Stochastic Gradient Descent (SGD)–based classifier performs intrusion detection. The SGD model is trained offline using labeled network traffic data and selected through comparative evaluation against multiple classifiers.

Due to its lightweight structure, fast convergence, and suitability for large-scale data processing, the SGD classifier supports efficient real-time inference even under high-throughput network conditions, making it appropriate for streaming intrusion detection systems. Based on the classification results, the system proceeds to the monitoring and visualization stage. Detection outcomes and network metrics are stored in InfluxDB, a time-series database optimized for streaming data. These records are visualized through Grafana dashboards, which provide real-time insights into traffic behavior, detection outcomes, and system performance. The visualization interface allows security administrators to observe traffic patterns, track potential threats, and respond quickly to suspicious activities.

Overall, the StreamGuard architecture transforms intrusion detection from a passive monitoring process into a dynamic and adaptive real-time defense mechanism capable of addressing the demands of modern high-speed network environments.

C. Model Comparison and Evaluation

To assess the effectiveness of the proposed StreamGuard framework, a comprehensive comparative evaluation was conducted using multiple machine learning classifiers under identical experimental conditions. The objective of this evaluation is not only to measure detection accuracy, but also to analyse system-level performance characteristics that are critical for real-time intrusion detection, such as inference latency, throughput, and generalisation capability.

In addition, the evaluation examines the stability of each model under high-volume traffic and its ability to maintain consistent performance during continuous streaming operation. By considering both classification effectiveness and execution efficiency, the analysis provides a balanced assessment of each model's suitability for deployment in real-time environments. This approach ensures that model selection is driven by practical deployment requirements rather than accuracy alone.

Experimental Setup-

All experiments were conducted using the preprocessed CSE–CIC–IDS2018 dataset, focusing on Denial-of-Service (DoS) attack scenarios along with benign network traffic. The dataset was divided into training and testing sets using an 80:20 split while maintaining the original class distribution. Feature scaling was applied prior to dimensionality reduction, and Principal Component Analysis (PCA) was used to retain approximately 95% of the original variance. The same transformed feature space was applied across all models to ensure a fair and consistent comparison during evaluation.

The classifiers evaluated in this study include Random Forest (RF), XGBoost (XGB), Gaussian Naïve Bayes (GNB), and Stochastic Gradient Descent (SGD). All models were trained offline using the training dataset and evaluated on the same test set. For real-time performance analysis, the trained models were integrated into the Kafka-based streaming pipeline, where network flow data is processed continuously. Prediction results and system metrics are stored in InfluxDB and visualized through Grafana dashboards, enabling monitoring of detection performance and system behavior in a real-time intrusion detection environment.

Evaluation Metrics-

The models were evaluated using both classification-level and system-level metrics. Classification performance was assessed using accuracy, precision, recall, F1-score, and receiver operating characteristic (ROC) analysis. To evaluate real-time suitability, inference latency, throughput, and generalization gap between training and testing performance were analyzed. These metrics provide a holistic view of each model's effectiveness in streaming intrusion detection scenarios.

Comparative Performance Analysis-

Ensemble-based models such as Random Forest and XGBoost demonstrated strong classification accuracy on the test dataset. However, these models exhibited higher inference latency and increased computational overhead, which negatively impacted throughput during real-time streaming. Gaussian Naive Bayes achieved fast inference speeds but showed comparatively weaker generalization performance, leading to reduced robustness under varying traffic conditions.

Real-Time Performance Evaluation-

To evaluate real-time suitability, each trained model was integrated into the Kafka-based streaming pipeline. Performance measurements revealed that SGD consistently maintained low-latency predictions and sustained high message throughput without causing bottlenecks in the stream processor. This behavior is particularly important for intrusion detection systems operating under strict time constraints, where delayed detection can lead to significant security risks.

Final Model Selection-

Based on the comparative analysis across classification metrics and real-time system performance, the Stochastic Gradient Descent classifier was selected as the final detection model for the StreamGuard framework. While ensemble models provided competitive accuracy, their higher computational cost limited scalability in streaming environments. The SGD classifier offered the most balanced and reliable performance, satisfying both detection effectiveness and real-time operational requirements.

RESULT

This section presents a detailed analysis of the experimental results obtained from the comparative evaluation of different machine learning models within the StreamGuard framework. The evaluation considers both classification performance and real-time processing efficiency, which are essential for intrusion detection in high-speed streaming environments. In addition to detection accuracy, metrics such as inference latency, throughput, and model generalization are analyzed to determine the most suitable model for real-time deployment. The experimental setup integrates the Kafka-based streaming pipeline with the trained models, while prediction results and system metrics are stored in InfluxDB and visualized using Grafana dashboards to support real-time monitoring and analysis.

Dimensionality Reduction using PCA-

Figure II
PCA Scree Plot

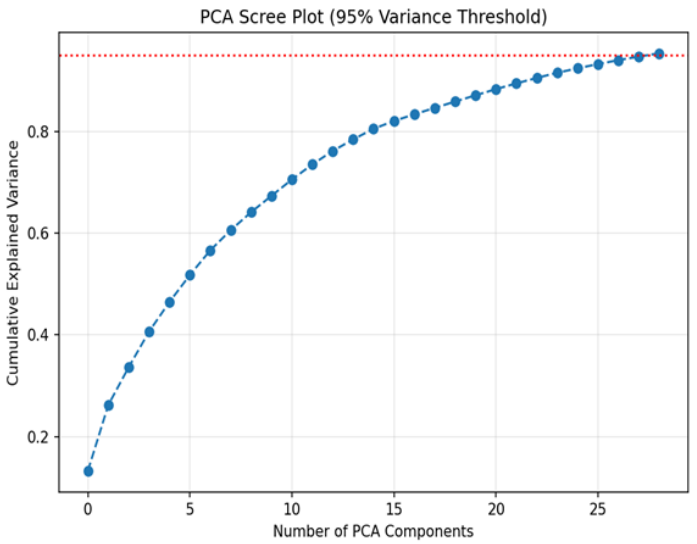


Figure II illustrates the PCA scree plot showing the cumulative explained variance with respect to the number of principal components. The curve indicates that the explained variance increases as more components are added. Approximately 95% of the total variance is retained when around 27–28 principal components are selected, as indicated by the threshold line. This demonstrates that PCA effectively reduces feature dimensionality while preserving most of the important information in the dataset.

Classification Performance Analysis-

Figure III
Training vs Testing Accuracy Comparison

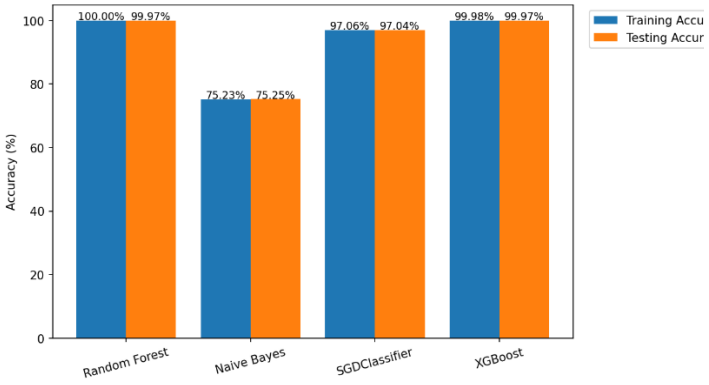
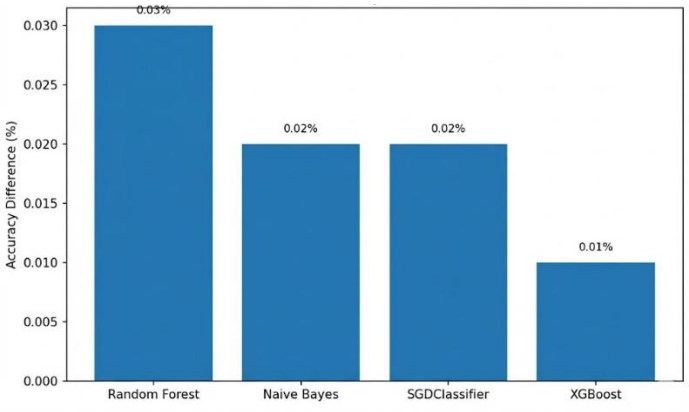


Figure III illustrates the comparison between training and testing accuracy for Random Forest, Naive Bayes, SGDClassifier, and XGBoost. Ensemble-based models such as Random Forest and XGBoost achieved very high classification accuracy, exceeding 99% on the test dataset. The SGDClassifier also demonstrated strong performance, achieving approximately 97% accuracy, while Naive Bayes showed comparatively lower accuracy

The small difference between training and testing accuracy across all models indicates stable learning behavior. However, accuracy alone does not fully capture the suitability of a model for real-time intrusion detection, particularly under streaming constraints.

Generalization Capability-

Figure IV
Generalization Gap



The generalization gap, defined as the difference between training and testing accuracy, is shown in Figure IV. Lower values indicate better robustness to unseen data. XGBoost exhibited the smallest generalization gap, followed closely by SGDClassifier and Naive Bayes. Random Forest showed a slightly higher gap, suggesting a greater tendency toward overfitting.

The consistently low generalization gap of the SGDClassifier demonstrates its ability to maintain reliable detection performance on unseen traffic, which is essential for dynamic and evolving network environments.

Inference Latency Evaluation-

Figure V
Inference Latency Evaluation

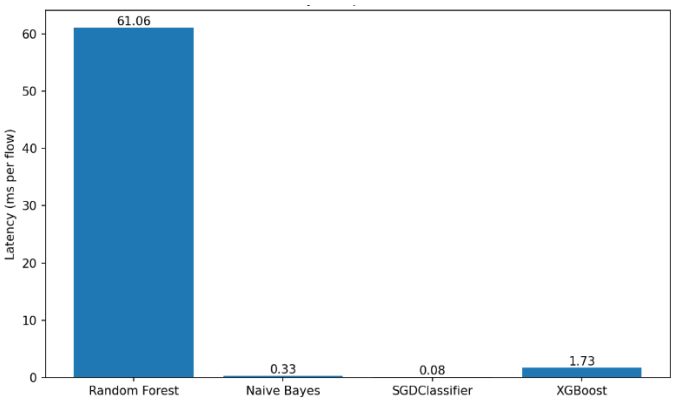
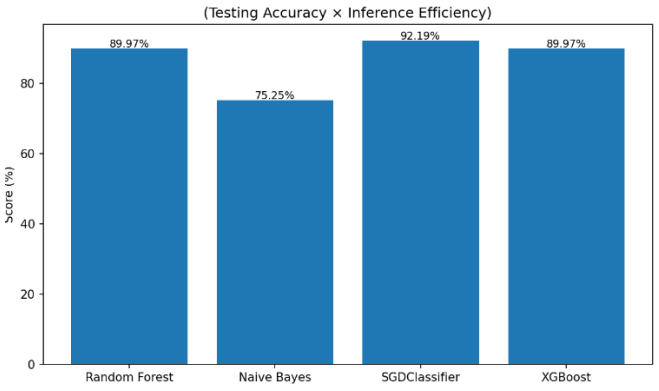


Figure V compares the inference latency of each model during real-time deployment. Random Forest incurred significantly higher latency, making it unsuitable for low-latency intrusion detection. XGBoost also exhibited noticeable inference delay due to its complex ensemble structure. Naive Bayes provided faster inference; however, the SGDClassifier achieved the lowest latency, demonstrating its effectiveness for real-time decision-making. Low inference latency is a critical requirement for streaming intrusion detection systems, as delayed predictions can reduce the effectiveness of timely threat mitigation.

Real-Time Suitability Assessment-

Figure VI
Real-Time Suitability Score



To quantify overall real-time effectiveness, a composite real-time suitability score was computed by combining testing accuracy and inference efficiency, as shown in Figure VI. The SGDClassifier achieved the highest suitability score among all evaluated models. Although Random Forest and XGBoost achieved high accuracy, their higher computational cost negatively impacted their suitability for real-time deployment. Naive Bayes, while computationally efficient, was limited by lower detection accuracy.

These results highlight that a balance between accuracy and efficiency is more important than maximizing classification accuracy alone in streaming intrusion detection scenarios.

Inference Throughput Comparison-

Figure VII
Inference Throughput Comparison

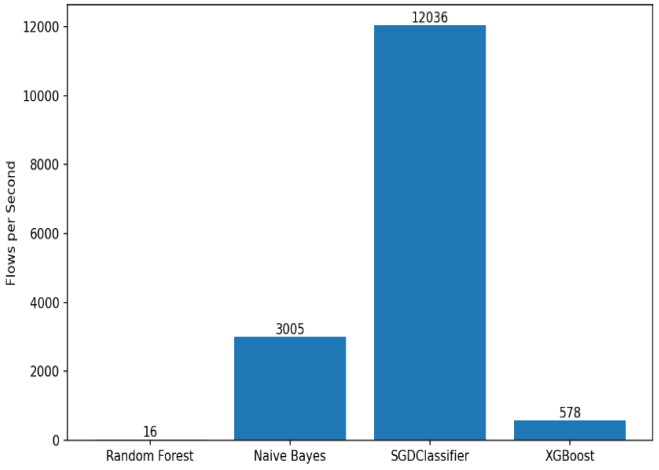


Figure VII illustrates the inference throughput comparison of different machine learning models used for intrusion detection. The results show that the SGD classifier achieves the highest throughput, processing approximately 12,036 flows per second. In comparison, Naïve Bayes handles around 3,005 flows per second, while XGBoost processes about 578 flows per second. Random Forest demonstrates the lowest throughput at approximately 16 flows per second due to its higher computational complexity. These results indicate that the SGD classifier is the most suitable model for real-time intrusion detection environments where high-speed processing of network traffic is required.

Discussion and Final Model Selection-

The experimental results demonstrate that while ensemble-based models such as Random Forest and XGBoost provide excellent classification accuracy, their higher inference latency and resource consumption limit their practicality in real-time streaming environments. Naive Bayes offers fast inference but lacks sufficient detection robustness.

The SGDClassifier emerges as the most balanced and effective model, achieving strong generalization, minimal inference latency, high throughput, and efficient resource utilization.

These characteristics make it the most suitable choice for deployment within the StreamGuard real-time intrusion detection framework.

Detailed Performance Analysis of the SGD Classifier-

After selecting the Stochastic Gradient Descent (SGD) classifier as the most suitable model for real-time intrusion detection, a detailed performance analysis was conducted to assess its detection reliability across different attack categories.

This analysis focuses on class-wise prediction behavior, threshold-independent evaluation, and robustness under class imbalance conditions.

Figure VIII
Confusion Matrix

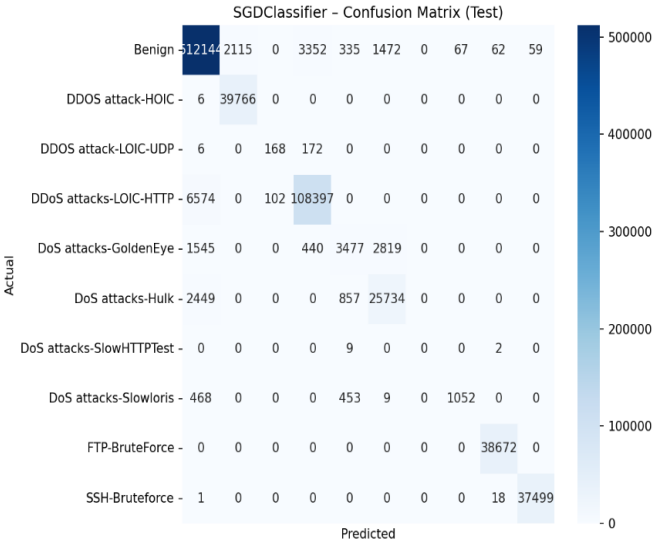
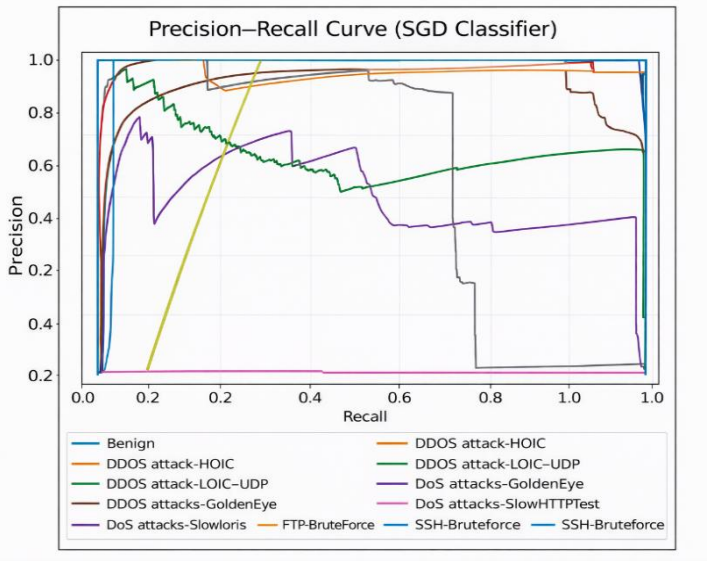


Figure VIII presents the confusion matrix of the SGD classifier on the test dataset. The results show a strong diagonal dominance across most traffic classes, indicating accurate classification of both benign traffic and multiple attack types. High detection accuracy is observed for high-volume attacks such as DDoS-LOIC-HTTP, DDoS-HOIC, and brute-force attacks, demonstrating the model's effectiveness in identifying dominant intrusion patterns. Minor misclassifications are primarily concentrated among lower-frequency attack categories, which is expected in large-scale, imbalanced intrusion detection datasets. Overall, the confusion matrix confirms that the SGD classifier maintains reliable class separation while minimizing false alarms.

Figure IX
Precision-Recall Curve



To further evaluate detection performance under varying decision thresholds, the Precision–Recall (PR) curves for individual traffic classes are shown in Figure IX. The PR analysis highlights consistently high precision values across a broad range of recall levels for most attack categories, indicating that the model can accurately detect malicious traffic while maintaining a low false positive rate. This behavior is particularly important for intrusion detection systems, where excessive false alerts can overwhelm security administrators. The observed variation among certain low-frequency classes reflects inherent class imbalance rather than model instability.

FIG X
Receiver Operating Characteristic (ROC) Curve

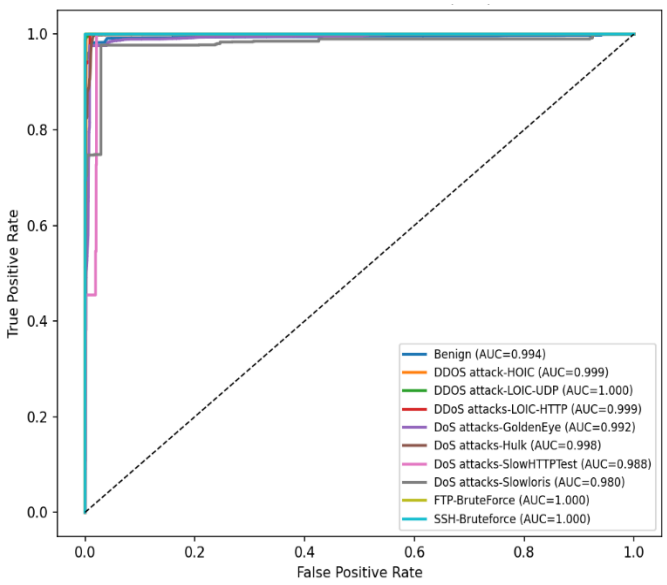
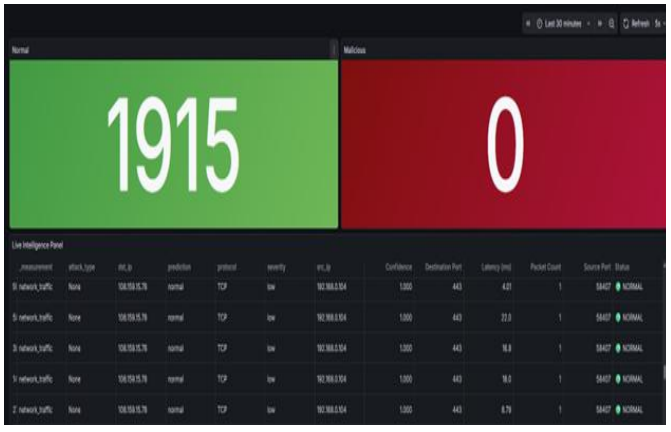


Figure X presents the ROC (Receiver Operating Characteristic) curves for the different attack classes detected by the model. The curves are positioned close to the top-left corner, indicating strong classification performance. The AUC values for most classes are very high, ranging from approximately 0.98 to 1.00, which demonstrates the model's ability to effectively distinguish between benign and malicious traffic. These results confirm that the classifier achieves high detection accuracy across multiple attack categories.

FIG XI
Real-Time Monitoring Dashboard



The real-time operational capability of the proposed system is demonstrated through the Grafana-based monitoring dashboard, as shown in Figure XI. The dashboard visualizes intrusion detection results and network flow metrics stored in InfluxDB, enabling continuous observation of system behavior. Key indicators such as the number of normal and malicious traffic instances, prediction confidence, latency, protocol information, and packet statistics are displayed in real time. The live intelligence panel provides detailed insights into each network flow, including source and destination information, classification outcomes, and severity levels. This visualization layer enhances system interpretability and allows security administrators to quickly identify suspicious activities and monitor network behavior effectively

CONCLUSION

This paper presented StreamGuard, a real-time intrusion detection framework that combines distributed stream processing with lightweight machine learning to address the limitations of conventional batch-based security approaches. By utilizing Apache Kafka for continuous data streaming and Principal Component Analysis (PCA) for feature dimensionality reduction, the proposed framework supports scalable and low-latency analysis of high-volume network traffic in dynamic environments. Experimental evaluation using the CSE–CIC–IDS2018 dataset demonstrated that StreamGuard achieves reliable detection performance while maintaining efficient real-time processing capabilities. A comparative assessment of several machine learning models indicated that the Stochastic Gradient Descent (SGD) classifier provides the most effective balance between detection accuracy, model generalization, inference latency, and processing throughput. Although ensemble models such as Random Forest and XGBoost achieved slightly higher accuracy, their computational requirements reduced their suitability for real-time streaming deployment. In contrast, the SGD-based approach showed consistent generalization ability, minimal inference delay, and high throughput, making it well suited for continuous intrusion detection scenarios.

The application of PCA significantly reduced the dimensionality of the feature space while preserving the most relevant traffic characteristics, thereby improving computational efficiency and enabling faster inference within the streaming pipeline. Furthermore, the Kafka-based architecture ensured reliable ingestion and processing of continuous network traffic streams. Detection results and system metrics were stored in InfluxDB and visualized through Grafana dashboards, enabling real-time monitoring of network behavior and intrusion detection outcomes. Overall, the StreamGuard framework demonstrates that integrating streaming architectures with efficient machine learning techniques can transform intrusion detection from a reactive offline task into a proactive real-time defense mechanism. The results indicate that such an approach provides a practical and scalable solution for securing modern enterprise and cloud-based network infrastructures.

REFERENCES

[1] Al-Hasan, M., Singh, A., & Banerjee, R. (2025). Improving intrusion detection systems by using deep learning methods on time series data. *Engineering, Technology & Applied Science Research (ETASR)*, 15(1), 94–102

[2] Rahman, A., Chen, Y., & Patel, V. (2025). Toward secure SDN infrastructure in smart cities: Kafka-enabled machine learning framework for anomaly detection. *Future Internet*, 17(9), 415.

[3] Wang, T., Li, J., & Chen, Z. (2025). CND-IDS: Continual novelty detection for intrusion detection systems. *arXiv preprint arXiv:2502.14094*.

[4] Abdul Wahid, & Hye-Young Kim. (2025). Making a real-time IoT network intrusion detection system (INIDS) using a realistic BoT–IoT dataset with multiple machine learning classifiers. *Applied Sciences*, 15(4), 2043.

[5] Jakotiya, K., Shirsath, V., & Inamadar, S. (2025). Intrusion detection using federated learning with neural networks. *Computer Science*, 26(2).

[6] Alhusseini, M. M., & Feizi Derakhshi, M. R. (2025). Hybrid AI-driven intrusion detection: Framework leveraging novel feature selection for enhanced network security. *arXiv preprint arXiv:2509.00896*.

[7] Gungor, O., Kale, I., Zhou, J., & Rosing, T. (2025). LIGHT-HIDS: A lightweight and effective machine learning-based framework for robust host intrusion detection. *arXiv preprint arXiv:2509.13464*.

[8] Zhou, L., Zhang, H., & Wang, Q. (2024). AE-integrated: Real-time network intrusion detection with Apache Kafka and autoencoder. *Future Internet*, 16(2), 115.

[9] Kumar, S., Lee, D., & Rahman, M. (2024). Machine learning- based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding, and feature extraction. *Journal of Big Data*, 11(5), 134.

[10] Aulia Novi, & Ryan Satria. (2024). Automated detection of network intrusions using machine learning in real-time systems. *International Journal of Computer Technology and Science*, 1(2), 20–23.

[11] Yellepeddi, S. M., Ravi, C. S., Vangoor, V. K. R., & Chitta, S. (2024). AI-powered intrusion detection systems: Real-world performance analysis. *Journal of AI-Assisted Scientific Discovery*, 4(1).

[12] Bhattacharya, S., Khanna, A., Ganapaneni, S., & Najana, M. (2024). Attention-based deep learning frameworks for network intrusion detection: An empirical study. *International Journal of Geographic Information Science*, September 2024.

[13] Elmoutaoukkil, A., Hamlich, M., Khatib, A., & Chriss, M. (2024). Network intrusion detection in big datasets using Spark environment and incremental learning. *IAES International Journal of Artificial Intelligence*, 13(4), 4414– 4421.

[14] Konda Srir Goud, M., Shivani, B. V. S., Selvi Reddy, C., Shravyasree, & Shreeya Reddy, J. (2024). An efficient real-time NIDS using machine learning methods. In *Cognitive Computing & Cyber-Physical Systems: 4th EAI International Conference (IC4S)*, Bhimavaram, India. Springer/EUDL.

[15] Malik, M. (2024). Real-time streaming architecture: Advanced insights and operational enhancements. *International Journal of Novel Research and Development*, 9(7).

[16] Mohammed, M. S., & Talib, H. A. (2024). Using machine learning algorithms in intrusion detection systems: A review. *Tikrit Journal of Pure Science*, 29(3), 63–74.

[17] Hartl, A., Vázquez, F. I., & Zseby, T. (2024). SDOoop: Capturing periodical patterns and out-of-phase anomalies in streaming data analysis. *arXiv preprint arXiv:2409.02973*.

[18] Richards, E. (2024). Deep learning techniques for intrusion detection systems: A comparative study of accuracy and efficiency. *Journal of AI-Assisted Scientific Discovery*, 4(2).

[19] Chen, K. (2024). Comparative analysis of machine learning methods in the detection of network intrusion. *Applied and Computational Engineering*, 106, 74–80.

[20] Al Lail, M., Garcia, A., & Olivo, S. (2023). Machine learning for network intrusion detection—A comparative study. *Future Internet*, 15(7), 243.

[21] Sirisha, G., Stephen, K. V. K., Suganya, R., Patra, J. P., & Lakshmi, T. R. V. (2023). An innovative intrusion detection system for high-density communication networks using artificial intelligence. *Engineering Proceedings*, 59(1), 78.

[22] Sukhadeo, B. S., Patil, R. N., Atole, R., Sinkar, Y. D., Patkar, U. C., & Chopade, R. (2023). MLIDS: A machine learning-based intrusion detection system using the NSL-KDD data. *International Journal of Intelligent Systems and Applications in Engineering*, 12(4s), 167–179.