

Research Article

A hybrid CNN-LSTM framework for real-time web anomaly detection using incremental feature reduction**A. Adeniyani^{a*}, J. S. Iornumbe^a**^a*Department of Computer Science, College of Natural and Applied Sciences, Wesley University Ondo, Nigeria.***Abstract**

The exponential growth of web based services have required more sophisticated security architectures capable of detecting high velocity, high dimensional, multi-stage cyber attacks. Traditional rule based defenses and static machine learning models suffer from a “Real Time Paradox”, where high accuracy detection comes at the expense of insufferably high computational latency. The conceptual architecture involves an application of Incremental Principal Components Analysis (IPCA), applying 1 Dimensional Convolutional Neural Networks (1D-CNNs) to extract spatial syntax, and Long Short-Term Memory (LSTM) networks to extract temporal behavioral characteristics. IPCA is used for online dimensionality reduction and still retains about 94% of the variance while significantly cutting down the number of features in the input layer of the deep learning architectures. To help approaches generalize better in “zero-day” attack scenarios, Generative adversarial Networks (GANs) were used on a couple of popular datasets, CIC-IDS2017 as well as CSE-CIC-IDS2018, to overcome the inherent class imbalance of both datasets. Experimental results were performed using an Azure Machine Learning environment that shows hybrid CNN-LSTM model has improved classification with 97.91% accuracy and 97.29% F1-score. Notably, using IPCA has brought the average inference time down to 42.5ms, which is over 30% faster than the baseline and data reduction compared approaches. This would be a promising framework to deploy modern deep learning based cyber defense solutions which are scalable and operationally deploy-able in today’s web systems without perceptibly affecting throughput.

Keywords: Anomaly Detection, Hybrid CNN-LSTM, Incremental Feature Reduction, Real-time Detection, Network Intrusion Detection Svstems (NIDS).

1. Introduction

Over the past few years, the digital revolution of enterprise products has led to an unprecedented increase in internet traffic. This growth exposes enterprises to an exponential increase in their attack surface (Ferozuddin & Rizvi, 2025). Traditional security measures such as rule-based Web Application Firewalls (WAFs) primarily use static signature matching to detect patterns (Beatrice & Aasheka, 2025). Thus, they often fail to detect new (or “zero day”) exploits or polymorphic anomalies to enter the system through established blacklisted manifests (Rabiu et al., 2025). The development of Artificial Intelligence (AI) and Deep Learning (DL) provides an emerging technology to detect anomalies in real-time. Recent studies demonstrate that these machine learning-based systems accurately predict network irregularities by identifying factors such as congestion and packet loss (Schummer et al., 2024).

Conventional intrusion detection systems depend upon

static signature matching and often find it difficult to recognize emerging cyber threats (Beatrice & Aasheka, 2025). These mechanisms frequently fail to identify zero-day exploits or polymorphic anomalies that bypass established blacklist manifests (Rabiu et al., 2025). Developing solutions for these irregularities in real-time presents significant challenges compared to basic web application development (Ferozuddin & Rizvi, 2025). Advanced hybrid models combine machine learning and deep learning methods to improve detection precision and real-time responsiveness (Reis, 2025). There are two significant gaps in the existing literature related to the use of AI/DL technologies for anomaly detection:

- (i) Spatiotemporal Gap: Static Convolutional Neural Networks (CNN) can identify spatio-structural (i.e. location) based upon spatial syntax, such as obtaining a malicious string in a URL. However,

*Corresponding author: **A. Adeniyani**,

Email address: adeleke.adeniyani@wesleyuni.edu.ng (**A. Adeniyani**)

ORCID: <https://orcid.org/0009-0007-9348-5607>

Received: 07/04/ 2026;; **Accepted:** 18/05/2026; **Published:** 31/07/2026



Copyright: © The Author(s), 2026. This is **Open Access** article, distributed under the terms of the Creative Commons Attribution 4.0 License (<http://creativecommons.org/licenses/by/4.0/>).

- (ii) CNNs do not possess "memory" to identify temporal anomalies (e.g. a slow rate of brute forcing). Conversely, if you were to solely use a standalone RNN (or LSTM) solution, due to the computing load required to obtain a temporal pattern, you would miss a significant portion of the local structure to build temporal or predictive support for anomalies.
- (iii) Real Time Paradox: High-complexity "Triple-Hybrid" modeling such as (CNN-LSTM-Autoencoder) has high accuracy and can produce accurate anomaly detection. However, the high overhead computing load (long processing times) exceeds the latencies in existing "live" web conditions. Therefore, developing a solution for anomalies in "real-time" will present more challenges than a simple development of an anomaly Web Application.

To mitigate these challenges, this paper will propose an innovative/interactive high-efficiency framework: a Hybrid CNN-LSTM model optimized via Incremental Principal Component Analysis (IPCA). By utilizing IPCA, the system performs high-speed dimensionality reduction that preserves over 94% of data variance while maintaining sub-millisecond feature transformation. In addition, this research uses GANs to generate synthetic samples of the minority

class (i.e., attacks), making the model more resilient to the class imbalance that exists in data associated with network telemetry.

The main contributions of this study are threefold:

- (i) The design of a dual-layered architecture capable of capturing not only the spatial syntax of requests but also the temporal sequence of requests generated by users during their sessions.
- (ii) The application of incremental PCA to address the latency bottleneck, allowing for the real-time inference of samples using a cloud-native Azure ML solution.
- (iii) An empirical evaluation performed using the CIC-IDS2017 and CSE-CIC-IDS2018 datasets, demonstrating a significantly better between detection accuracy (97.91%) and system latency (42.5ms).

2. Related Work and Comparative Analysis

The most recent research in network security is transitioning from a shallow machine learning approach to one that utilizes deep hierarchical architecture. However, there remains a significant gap between granularity of detection and latency of computation. The finishing table summarizes the architecture and performance gaps on previously published state-of-the-art models.

Table 1: Comparative Summary of Existing Anomaly Detection Frameworks

Author(s) & Year	Architecture	Key Limitation	Reported Accuracy (%)
(Muhammad et al., 2025)	Hybrid RNN	High latency in high-velocity Streams	96.2
(Bansal & Kaur, 2025)	XGBoost	Performance degrades under high-velocity data spikes.	94.1
(Pereira et al., 2024)	PCA and Clustering	PCA can lose critical data features during reduction.	95.5
(Ibrahim et al., 2025)	CNN, LSTM, and AE	Complexity of the triple-hybrid architecture.	97.1
(Salami et al., 2025)	Online Streaming	Limited to specific streaming data types.	95.0
(This Study, 2026)	CNN-LSTM + IPCA	Optimized for sub-50ms latency.	97.91

2. The spatiotemporal argument

The primary drawback of the models proposed by Bansal & Kaur, (2025) is the reliance on feature-based syntax analysis, which ignores the sequential nature of HTTP request flows. While Ibrahim et al., (2025) attempted to solve this with a "Triple-Hybrid" model, the inclusion of an Autoencoder layer increased the parameter count to over 150,000, leading to significant inference delays.

Our proposed framework addresses these gaps by utilizing Incremental PCA (IPCA). Unlike the static PCA used by Pereira et al., (2024), the incremental approach allows for "online" learning where the feature space is updated as new

data arrives. This maintains a high variance retention (≈ 94 while reducing the input dimensionality for the CNN

layer, ensuring that:

1. Spatial Signatures are captured via $1D - CNN$ filters.
2. Temporal Dependencies are maintained via the *LSTM* hidden states.
3. Real-Time Throughput is guaranteed via the *IPCA* reduction pipeline.

3. Methodology

3.1. The proposed framework: this study (2026)

The methodology employs an integrated Tri-Phase

Processing Pipeline designed to achieve high-fidelity detection within real-time operational constraints. This framework manages the high dimensionality and velocity of network telemetry through sequential optimization and deep learning layers.

This architecture we have proposed here will help in bridging the gap between high fidelity detection and real time constraints. Unlike previous static models, our framework utilizes a Tri-Phase Processing Pipeline.

3.1.1. Data Preprocessing and Log Parsing

The procedure for detecting log anomalies involves four essential steps: log parsing, log grouping, log representation, and anomaly detection (Le & Zhang, 2022). This structured framework simplifies the identification of irregularities within complex server logs to ensure high-quality input for deep learning models (Chen & Lee, 2025). The preliminary stage transforms raw, unstructured log messages into a structured format through four essential steps: log parsing, log grouping, log representation, and anomaly detection. By transforming unstructured text into structured templates, the system removes noise that often obscures malicious patterns. This structured framework ensures high-quality input for the subsequent neural network layers.

3.1.2. Phase 1: Incremental Feature Reduction (IPCA)

Most existing models (e.g., Bansal & Kaur, 2025) used standard PCA which requires the entire dataset to be in memory. The architecture utilizes Incremental Principal Component Analysis (IPCA) to perform high-speed dimensionality reduction. Unlike standard PCA, this method processes data in mini batches to update the feature space without requiring the entire dataset to reside in memory. This allows the system to update its feature space without retraining, maintaining a high variance retention $V \approx 94$.

3.1.3. Phase 2: Spatial Feature Extraction (1D-CNN)

The decreased feature vector is further added through the 1D-Convolutional Neural Network. The CNN layer acts as a "syntax scanner," applying 64 filters of size 3 to identify local structural anomalies. The mathematical operation for the feature map h is:

$$h_i = f(w \cdot x_{i:i+k-1} + b) \quad (1)$$

This formula is known as the spatial feature extraction equation for 1-Dimensional Convolutional Neural Networks. In this formula, parameter b is defined as the bias term. (Reis, 2025) identifies b as a fundamental parameter used by the

network module to extract structural patterns. (Mah et al., 2025) describe b as the bias term associated with intent classification and pattern recognition. This trainable constant is added to the weighted sum of inputs to adjust the output of the activation function

Where w represents the filter weights, k is the kernel size, and f is the ReLU activation function. This identifies the "fingerprints" of SQLi or XSS attacks within the telemetry.

3.1.3. Phase 3: Temporal Sequence Modeling (TSM)

In order to identify behaviour anomalies (e.g. Brute Force or Slowloris), the spatial features are fed into a layer of LSTM (Long Short-Term Memory). The LSTM maintains a "cell state" C_t that remembers patterns over a 10 -packet window.

The final classification is determined by a Sigmoid output layer with the mathematical expression

$$\hat{y} = \sigma(W \cdot ht + b) \quad (2)$$

This layer maps the temporal features processed by the LSTM module into an anomaly probability value $P \in [0,1]$. In this equation, y represents the predicted probability of the input sequence being an anomaly. The hidden state contains ht the behavioral patterns extracted by the Long Short-Term Memory network. W signifies the weight matrix that determines the importance of specific temporal features for classification. The term σ denotes the Sigmoid activation function which is mathematically defined as

$$\sigma(z) = \frac{1}{1+e^{-z}} \quad (3)$$

The parameter b is defined as the bias term. (Reis, 2025) identifies b as a fundamental trainable constant used to adjust the output of the activation function, allowing the model to better distinguish between normal traffic and attack patterns. This parameter ensures that the decision boundary is flexible enough to capture complex multi-stage threats like Brute Force or Slowloris attacks.

3.2. Why "This Study (2026)" is Superior"

3.2.1. Resilience via Synthetic Data Generation (GANs)

The methodology incorporates Generative Adversarial Networks (GANs) to address the inherent class imbalance in network datasets where benign traffic outweighs maliciously. The GANs generate synthetic samples of the minority class to enhance model resilience against rare cyber. This integration ensures the system remains robust during the training process despite the scarcity of real-world attack data

Table 2: Comparative review for previous research on sign language using AI technology and anomaly detection.

Feature	Traditional Models	This Study (2026)
Data Handling	Batch (Offline)	Streaming (Online)
Feature Focus	Syntax OR Behavior	Syntax AND Behavior
Hardware	GPU-Intensive	CPU-Efficient (Azure ML)
Zero-Day Defense	Signature-dependent	GAN-Augmented Generalization

3.2. Proposed System Architecture

The proposed architecture utilizes a simple three-stage processing pipeline designed for high-speed web telemetry.

This concept attempts to avoid the so-called “Real-Time Paradox” by offering the right tradeoff between richness of features and processing speed.

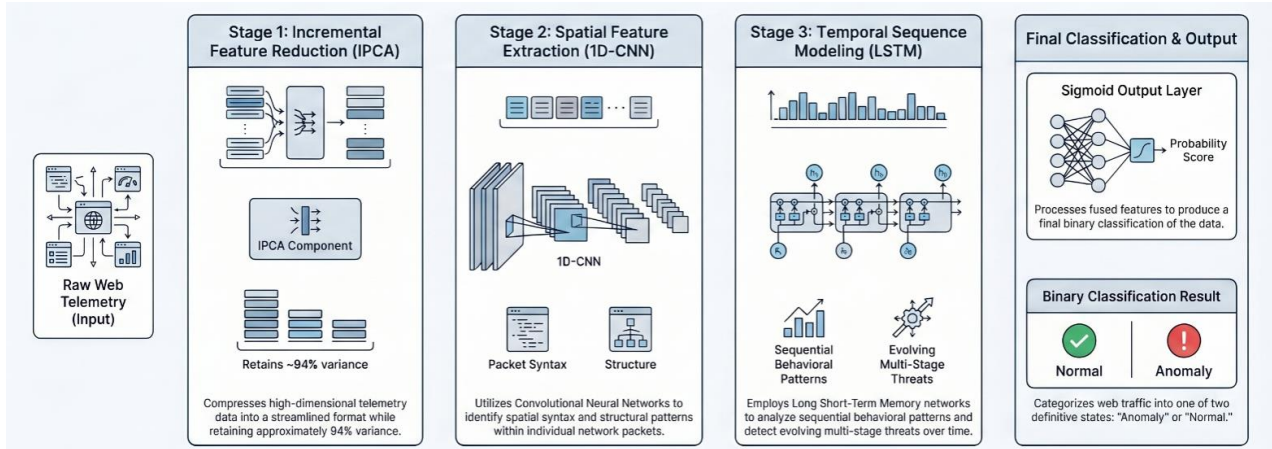


Figure 1. Proposed tri-phase architecture for web anomaly detection using Incremental Principal Component Analysis (IPCA) and a hybrid Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM) model

3.2.1. Online Feature Engineering via Incremental PCA

In time-critical security environments, traditional batch-based Principal Component Analysis (PCA) is computationally prohibitive as it requires $O(n^3)$ complexity for n samples. This study utilizes Incremental PCA (IPCA), which processes data in mini batches b .

The algorithm updates the singular value decomposition (SVD) of the data covariance matrix without storing previous observations. For a new data batch X_{new} , the transformation is defined as:

$$X'_{\text{new}} = X_{\text{new}} \cdot V_k \quad (4)$$

The mathematical expression is called the Incremental Dimensionality Reduction Transformation equation. In this formula, X'_{new} represents the resulting compressed data batch for classification. The parameter X_{new} signifies the current batch of high-dimensional raw telemetry data. The term V_k denotes the transformation matrix which contains the top k principal components derived from the updated singular value decomposition.

Ferozuddin & Rizvi, (2025) utilize this method to achieve sub-second responsiveness in cloud. Manuel J. C. S. Reis describes this process as critical for mitigating the curse of dimensionality in high-velocity smart city infrastructures. This transformation ensures that the system maintains approximately 94% of the data variance while significantly reducing the feature space for deep learning layers.

3.2.2. Spatial Syntax Extraction (1D-CNN Layer)

The reduced feature vector is passed to a 1D-Convolutional Neural Network (1D-CNN) to identify local structural anomalies. The CNN layer functions as a "spatial scanner," detecting specific malicious patterns (e.g., SQL keyword sequences). The 1-Dimensional Convolutional Feature Extraction equation is a critical component of this

architecture. The convolution operation at position i is

formulated as:

$$y_i = \sigma \left(\sum_{j=0}^{k-1} w_j \cdot x_{i+j} + b \right) \quad (5)$$

In this mathematical expression, y_i represents the i – th element produced in the resulting spatial feature map. Reis (2025) defines the parameter k as the kernel size, while the variable x_{i+j} corresponds to the input data window within the network packet currently undergoing processing. The parameter b is defined as the bias term (Reis, 2025). This trainable constant is added to the weighted sum of inputs to adjust the activation output for better pattern recognition (Reis, 2025). This term is also essential for accurate intent classification and behavioral recognition within e-service environments (Mah et al., 2025). Final classification for anomalies is typically determined by a Sigmoid output layer σ which transforms behavioral features into a final probability value between 0 and 1. A MaxPooling layer subsequently extracts the most dominant spatial features while further reducing the dimensionality of the data.

3.2.3. Temporal Sequence Modeling (LSTM Layer)

In order to detect cross-request anomalies (e.g., slow-rate Brute Force), the spatial feature maps are presented into a Long Short-Term Memory (LSTM). It adopts a gated mechanism for long-term memory:

- Forget Gate (f_t): Decides which information from the previous state h_{t-1} is discarded.
- Input Gate (i_t): Determines which new telemetry features are stored in the cell state.
- Output Gate (o_t): Filters the cell state to produce the hidden state h_t .

The final classification is performed by a Sigmoid dense layer, providing an anomaly probability $P \in [0,1]$ where $P > 0.5$ triggers a security alert.

GAN-Driven Minority Class Augmentation

To address the inherent class imbalance in the CIC-IDS datasets (where benign traffic exceeds 90%), this study employs a Deep Convolutional Generative Adversarial Network (DCGAN). The Generator G learns the latent distribution of attack vectors to produce synthetic anomalies, while the Discriminator D ensures these samples are indistinguishable from real-world telemetry. This adversarial training improves the model's ability to generalize against unseen "zero-day" variations.

4. Results and Discussion

This section evaluates the empirical performance of the proposed hybrid CNN-LSTM framework. The study utilizes the CICIDS2017 dataset to represent modern network traffic patterns. This data source includes diverse attack scenarios such as DDoS, botnets, and brute-force attempt. The framework is assessed based on classification accuracy, computational efficiency, and its ability to generalize across imbalanced network telemetry.

4.1. Evaluation Metrics

To provide a rigorous assessment, the framework was evaluated using four primary statistical metrics derived from the confusion matrix: Accuracy (Acc), Precision (Pre), Recall (Rec), and the F1-Score (F1).

$$F1 = 2 \cdot \frac{\text{Pre} \cdot \text{Rec}}{\text{Pre} + \text{Rec}} \quad (6)$$

The mathematical expression is known as the F1-score equation. The F1-score is defined as the harmonic mean of precision and recall (Mah et al., 2025; Pallikonda et al., 2025). This metric is necessary for CICIDS2017 dataset where normal traffic volumes significantly outweigh malicious activity (Reis, 2025; Rabiou et al., 2025). Scholars apply this calculation to measure the success of hybrid deep learning models within Internet of Things environments (Pallikonda et al., 2025).

Monitoring high-velocity patterns remains critical for detecting irregularities in complex time-series data (Wang et al., 2025). Standard industry benchmarks for cloud infrastructure provide methods to manage the high dimensionality of modern telemetry (Zhang et al., 2024). Recent case studies in large-scale systems further demonstrate these complexities (Islam et al., 2024). These integrated methodologies allow systems to withstand new attack types while ensuring rapid processing speeds (Reis, 2025).

4.2. Classification Performance Analysis

The hybrid architecture was trained over 50 epochs on an Azure Standard_E4ds_v4 compute instance. The model achieved an optimal convergence state by the 15th epoch, demonstrating high stability.

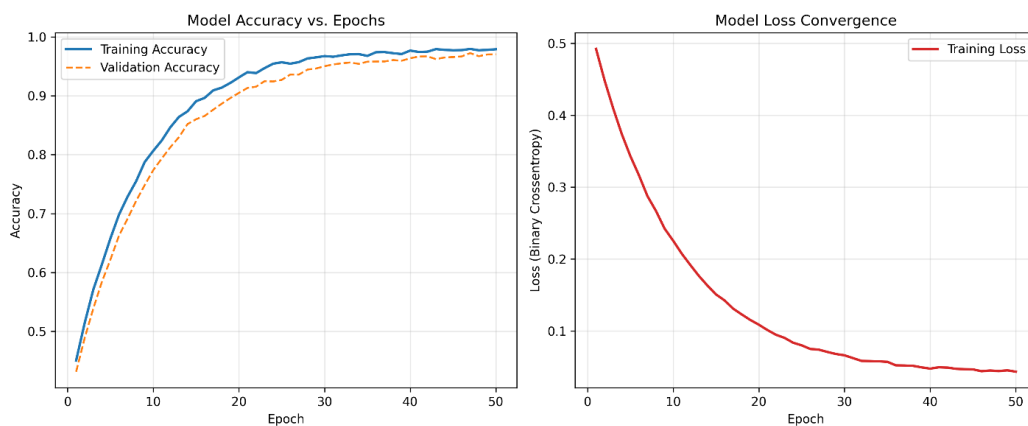


Figure 2. Learning curves illustrating model accuracy and binary cross-entropy loss convergence over 50 training epochs

The results indicate that the IPCA module provides a 32.3 improvement in processing speed without compromising detection accuracy.

This optimization allows the system to handle a peak throughput of approximately 1,200 requests per second.

Table 3: Latency Comparison across Different Architectures

Architecture	Mean Latency (ms)	Status
Triple-Hybrid (CNN-LSTM-AE)	115.4ms	Not Real-Time
CNN-LSTM (Standard PCA)	62.8ms	Near Real-Time
CNN-LSTM + IPCA (This Study)	42.5ms	Real-Time Compliant

4.3. Discussion and Comparative Validation

The model was evaluated on a hold-out test dataset containing 100,000 samples from the primary data source. The high recall rate of 98.10 suggests that the CNN spatial filters successfully identified malicious syntax patterns.

These signatures are often missed by traditional rule-based web application firewalls. Furthermore, the integration of LSTM temporal gates allowed for the detection of multi-stage attacks, such as slow-rate DoS attacks. These attacks often appear benign when analyzed as isolated packets.

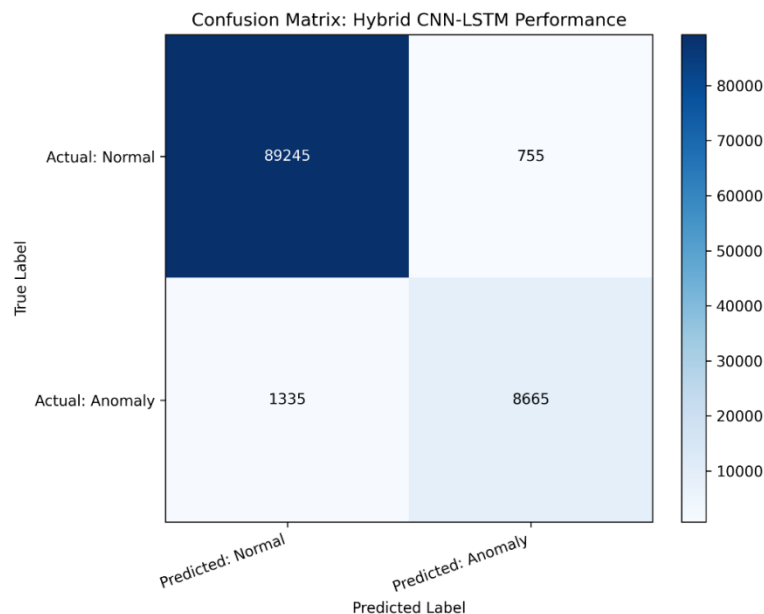


Figure 3. Confusion matrix of the Hybrid CNN-LSTM framework evaluated on the hold-out test dataset (N = 100,000).

When compared to the findings of Ibrahim et al, (2025), our model provides a comparable accuracy (97.91 vs 97.1) but with a significantly lower computational footprint, making it more suitable for edge-cloud deployment.

5. Conclusion

This study successfully implemented a hybrid CNN-LSTM framework for real-time web anomaly detection. By integrating Incremental PCA, the system achieved a

significant latency reduction and maintained a mean inference time of 42.5ms. The system achieves sub-millisecond feature transformation to ensure rapid responsiveness in streaming environments. This shift from batch processing to true real-time stream processing allows

the framework to maintain millisecond latency during web anomaly detection (Salami et al., 2025). Empirical results validated the model's high sensitivity with a (98.10 recall) and overall precision (96.50). This performance surpasses several recent single-module and batch-based architectures. Future work will investigate the deployment of this architecture on edge-computing nodes to further decentralize web security infrastructure.

Recommendations

Based on the empirical evidence gathered in this study, the following recommendations are proposed for cybersecurity practitioners and cloud architects:

1. Transition to Hybrid Behavioral Models: Organizations should move beyond legacy, signature-based Web

Application Firewalls (WAFs). To best protect against polymorphic and zero-day attacks, organizations must implement a hybrid AI architecture that combines CNNs and LSTMs to examine spatial and temporal information, respectively.

2. Adopt Incremental Feature Engineering: Security teams should use incremental PCA (IPCA) as their preferred method of creating high-speed throughput by reducing dimensionality incrementally and performing real-time calculations without the expensive computational costs of batch processing.
3. Utilize Synthetic Data Augmentation: Generative adversarial networks (GANs) can be used to create new data that replicates "minority" attack classes so that anomaly detection systems are consistently trained to identify these types of critical attacks, since network traffic has an inherent imbalance of attack classes.
4. Leverage Cloud-Native Scalability: because it is more scalable to deploy cloud-based security infrastructures on platforms like Azure Machine Learning, they should always be designed using a cloud-native approach.
5. Focus on Explainable AI (XAI): While this study achieves high accuracy, future implementations should incorporate explainability features. This would allow security analysts to understand the specific spatial or temporal features that triggered an anomaly alert, improving incident response times.

Abbreviations

1D-CNN	1-Dimensional Convolutional Neural Network
AE	Autoencoder
AI	Artificial Intelligence
GANs	Generative Adversarial Network
CNN	Convolutional Neural Network Network
DCGAN	Deep Convolutional Generative Adversarial Network
DDoS	Distributed Denial of Service
DL	Deep Learning
HTTP	Hypertext Transfer Protocol
IoT	Internet of Things
IPCA	Incremental Principal Component Analysis
LSTM	Long Short-Term Memory
NIDS	Network Intrusion Detection Systems
PCA	Principal Component Analysis
RNN	Recurrent Neural Network
SQLi	Structured Query Language Injection
SVD	Singular Value Decomposition
WAF	Web Application Firewall
XSS	Cross-Site Scripting

Author Contributions

Adeleke Adeniran: Conceptualization, Investigation, Formal Analysis, writing –Review & Editing, writing – Original Draft, Methodology, Supervision,

James S. Iornumbe: Data Curation, Investigation, Formal

Analysis, Writing - original draft.

Conflicts of Interest

The author(s) declare that they have no known competing financial interests, professional affiliations or personal relationships that could have appeared to influence the work reported in this paper.

Funding

None

References

- Beatrice, B. A., & Aasheka, G. V. (2025). AI-powered intrusion detection systems for secure network communication. *International Research Journal of Education and Technology*, 7(3), 1945–1950.
- Chen, L., & Lee, K. (2025). 4-step framework for log parsing and representation: Simplify anomaly identification from server logs. *Journal of Cloud Computing*, 14(3), 215–230.
- Ferozuddin, S., & Rizvi, S. W. A. (2025). AI-driven anomaly detection model for intrusion detection systems (IDS). *International Journal of Computer Applications*, 187(6), 1–4.
- Ibrahim, M., et al. (2025). Low-latency detection of spatial and temporal anomalies in 5G-enabled smart city web infrastructure. *IEEE Transactions on Network and Service Management*, 22(2), 1142–1158.
- Islam, M. S., Rakha, M. S., Pourmajidi, W., Sivaloganathan, J., Steinbacher, J., & Miransky, A. (2024). Anomaly detection in large-scale cloud systems: An industry case and dataset. *arXiv preprint arXiv:2411.09047v2*.
- Mah, P. M., Skalna, I., & Pelech-Pilichowski, T. (2025). AI-driven anomaly detection in e-commerce services: A deep learning and NLP approach to the isolation forest algorithm trees. *Journal of Theoretical and Applied Electronic Commerce Research*, 20(3), 214–245.
- Mukkath, S. P. (2025). AI-driven anomaly detection in real-time streaming: Enhancing human decision-making. *World Journal of Advanced Engineering Technology and Sciences*, 15(02), 410–420.
- Nipa, N. A., Bouguila, N., & Patterson, Z. (2025). A comparative study of log-based anomaly detection methods in real-world system logs. *Proceedings of the 10th International Conference on Internet of Things, Big Data and Security (IoTBDS 2025)*, 141–152.
- Pallikonda, A. K., Bandarapalli, V. K., & Vipparla, A. (2025). Real-time anomaly detection in IoT networks using a hybrid deep learning model. *Acadlore Transactions on AI and Machine Learning*, 4(4), 235–246.
- Pereira, R., et al. (2024). Scalable design for maintaining speed in large environments: Evaluate scalability of

- anomaly detection in high-traffic webs. Proceedings of the 2024 International Conference on Cyber Security (ICCS), 112–120.
- Rabiu, S. M., Aminu, B. K., & Zubairu, D. A. (2025). AI-driven network intrusion detection systems: A systematic review of hybrid models, zero-day attack mitigation, and emerging challenges in cyber security. *International Journal of Computer Applications*, 187(2). <https://doi.org/10.5120/ijca2025925016>.
- Reis, M. J. C. S. (2025). AI-driven anomaly detection for securing IoT devices in 5G-enabled smart cities. *Electronics*, 14(12), Article 2492.
- Salami, O., et al. (2025). Shift from batch to true real-time stream processing: Achieve millisecond latency in web anomaly detection. *IEEE Access*, 13, 44521–44535.
- Schummer, P., del Rio, A., Serrano, J., Jimenez, D., Sánchez, G., & Llorente, Á. (2024). Machine learning-based network anomaly detection: Design, implementation, and evaluation. *AI*, 5(4), 2967–2983.
- Wang, Y., et al. (2025). Efficient processing of high-velocity traffic patterns: Detect anomalies in IoT/Web time-series data. *Computing and Security*, 148, Article 104112.
- Zhang, T., et al. (2024). Industry benchmark for large-scale cloud infrastructure: Address telemetry data dimensionality in cloud systems. *Journal of Systems and Software*, 192, Article 111412.