FUDMA Journal of Sciences (FJS)
ISSN online: 2616-1370
ISSN print: 2645 - 2944
Vol. 10 No. 12, June, 2026, pp 339 – 348
DOI: https://doi.org/10.33003/fjs-2026-1012-5375

Adaptive Hybrid Random Forest–Lstm Framework for Network Anomaly Detection in Dynamic Cloud Environments

***Tamakloe Vivian Anuoluwa and Jam Donald Terdoo**

Federal University Dutsin-Ma, Katsina State, Nigeria.

*Corresponding authors' email: v.tamakloe@fudutsinma.edu.ng

ABSTRACT
The increasing adoption of cloud computing introduces security challenges due to dynamic network traffic. Traditional intrusion detection systems and single-model machine learning (ML) approaches struggle to detect sophisticated cyberattacks and adapt to changing patterns. This study proposes an Adaptive Hybrid Random Forest (RF) - LSTM Framework for Network Anomaly Detection in Dynamic Cloud Environments, designed to leverage the complementary strengths of ML and deep learning (DL) techniques for enhanced anomaly detection performance. Using the CICIDS2017 dataset, preprocessing included data cleaning, feature encoding, normalization, feature selection, and sequence generation. RF handled feature importance ranking and classification, while LSTM captured temporal dependencies. Their outputs were integrated via a weighted fusion mechanism with an adaptive strategy that dynamically adjusts model contributions based on prediction performance.
Experimental results showed RF achieved 99.99% precision; 99.98% across accuracy, recall, F1-score, and ROC-AUC. LSTM achieved 99.84% accuracy, 99.90% precision, 99.73% recall, 99.82% F1-score, and 99.83% ROC-AUC. The hybrid framework further improved performance to 99.96% across accuracy, precision, and ROC-AUC, 99.94% recall, and 99.95% F1-score. ROC analysis showed near-perfect class discrimination, and the adaptive weighting responded well to changing traffic. Statistical validation (Shapiro–Wilk and Wilcoxon signed-rank tests) confirmed that improvements were significant. The results show that the suggested Adaptive Hybrid Framework successfully integrates temporal sequence modeling, adaptive decision fusion, and feature-based learning to produce extremely reliable and accurate anomaly detection in dynamic cloud environments. By increasing detection reliability while preserving flexibility in response to changing cyberthreats, the framework provides a viable way to improve cloud security.

Received: 26th June 2026
Accepted: 11th July 2026
Published: 5th August 2026

Keywords: Adaptive Mechanism, Cloud Computing, Hybrid Random Forest – LSTM, Intrusion Detection System, Network Anomaly Detection

INTRODUCTION

Cloud computing's quick development has drastically changed how data is processed, stored, and delivered. Cloud environments offer scalable, adaptable, and affordable infrastructure that supports a variety of applications in industries such as corporate systems, healthcare, finance, and education (Tabrizchi & Kuchaki-Rafsanjani, 2020). Notwithstanding these benefits, the extensive use of cloud computing has brought about significant security issues, especially in the area of network security (Alouffi et al., 2021). Cloud infrastructures are inherently dynamic, characterized by virtualization, multi-tenancy, elastic resource allocation, and distributed architectures. Due to these characteristics, network traffic patterns are extremely complicated and changeable, making it challenging to discern between malicious and legitimate activity (Mishra et al., 2022). Cloud environments are more vulnerable to a variety of cyberthreats, such as denial-of-service (DoS) assaults, probing, data exfiltration, and unauthorized access (Kumar et al., 2024). As a result, real-time anomaly detection has become essential to preserving cloud-based systems' availability, confidentiality, and integrity (Mishra et al., 2022).

Traditional methods of detecting network anomalies, especially signature-based and rule-based systems, depend on predetermined patterns of known threats that have already been discovered, but they are not very good at spotting new or changing attack patterns, which are prevalent in contemporary cloud systems. As a result, intelligent, data-driven strategies based on deep learning (DL) and machine learning (ML) are becoming more popular (Tufail et al., 2021). Due to their resilience, interpretability, and capacity to handle high-dimensional datasets, ML techniques like Random Forest (RF) have been extensively used for anomaly detection (Abdulhammed et al., 2022). RF models are good at classifying structured data and finding pertinent features. But they do not specifically take temporal dependencies in network traffic into account and are intrinsically static (Lo et al., 2023).
In contrast, Long Short-Term Memory (LSTM) networks and other DL models are made to recognize temporal and sequential patterns in data. LSTM models learn relationships over sequences of network events and have demonstrated promising results in identifying abnormalities that change over time (Imrana et al., 2021). However, in comparison to conventional ML methods, these models may be less

interpretable and frequently demand substantial computational resources (Khan et al., 2023).

In light of these constraints, recent studies have investigated hybrid strategies that blend DL and classical ML models to take advantage of their complementary advantages (Mayuranathan et al., 2022). Demonstrated how a hybrid deep-learning pipeline can reduce overfitting, poor classification accuracy, and high false-positive rates in cloud IDS using the DARPA dataset. The diversity, drift, and changing attack patterns of contemporary dynamic cloud traffic are not fully captured by DARPA because it is an older benchmark. In comparison to traditional IDS baselines, Modi (2025) employed NSL-KDD and CICIDS2017 for a cloud-edge CNN-LSTM system that enhanced detection accuracy, precision, recall, and false-alarm reduction. Eguavoen et al. (2025) proposed a two-stage hybrid intrusion detection framework combining CNN, LSTM, and AdaBoost for classifying attacks in IoT networks. The framework achieved 99.70% accuracy, 99.90% precision, 99.78% recall, 99.84% F1-score, and a low false positive rate of 2.43%, outperforming prior conventional IDS benchmarks but model's practical viability in live, resource-constrained IoT environments remained unvalidated. Also, Sajid et al. (2024) achieved strong binary and multiclass results across four benchmarks using a layered hybrid approach that combines XGBoost and CNN for feature selection with LSTM classification using CICIDS2017, UNSW-NB15, NSL-KDD, and WSN-DS. Temporal drift, operational latency, and real-time cloud deployment are still problems because the framework is still computationally layered and benchmark-heavy.

The need for a reliable, adaptive, and computationally efficient anomaly detection framework that: combines the strengths of both classical and DL models to captures both static and temporal features of network traffic; and adapts to evolving patterns in cloud environments is justified by these studies' recurrent gaps: limited validation in live cloud environments, class imbalance, outdated or redundant data, and not incorporating adaptive mechanisms to maintain performance over time in dynamic cloud environments. Because LSTM provides temporal learning for changing attack sequences and idea drift, while RF provides robustness and good feature discrimination on structured data, this study proposes an adaptive hybrid RF–LSTM framework.

MATERIALS AND METHODS

The method adopted in this study follows the experimental research design procedure: data collection, preprocessing, development, training, and evaluation of models on network traffic datasets with emphasis on supervised learning, sequence modeling, feature selection, and statistical validation of results. Figure 1 presents a high-level view of the method's proposed system architecture.

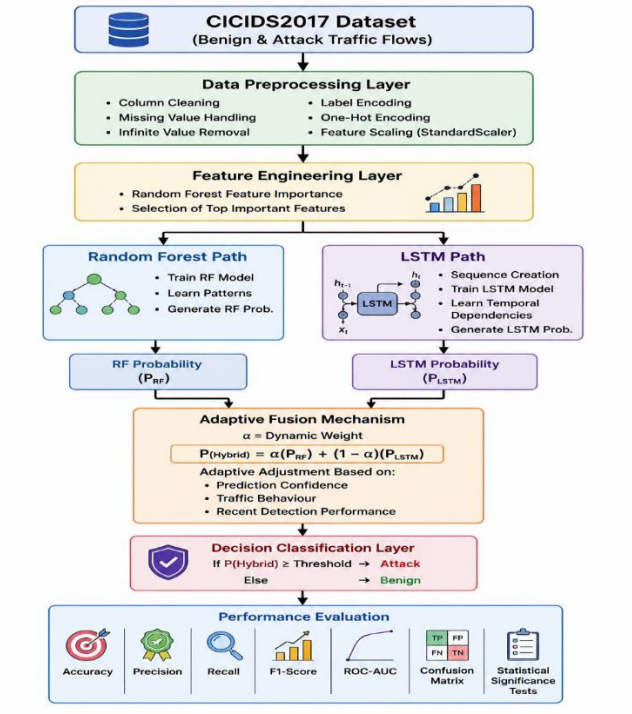

Figure 1: Proposed Adaptive Hybrid Random Forest–LSTM Framework

Data Source

The dataset used in this study is a publicly available network intrusion detection dataset derived from the CICIDS2017 collection (https://www.kaggle.com/datasets/naeem41/cicids2017-dataset). The dataset contains labeled network traffic records representing both benign and malicious flows, making it suitable for supervised anomaly detection. For this study, the combined CSV version of the dataset was used to simplify loading, preprocessing, and experimental analysis. The dataset includes diverse network behavior patterns that reflect real-world traffic conditions, thereby making it appropriate for investigating anomaly detection in dynamic cloud environments. The choice of dataset was informed by the need for a realistic benchmark that supports both binary classification and temporal analysis. To distinguish benign from malicious traffic, the labels were converted into two classes: benign traffic, assigned the value 0, and attack traffic, assigned the value 1. Ethical considerations were observed throughout the study. The dataset used is publicly available

and does not contain personally identifiable information in a form that requires direct user consent for academic use. No attempt was made to identify individuals or reconstruct private communications from the traffic records. The dataset was used strictly for academic and research purposes, to improve cybersecurity detection systems rather than to enable offensive use. All preprocessing, analysis, and reporting were conducted transparently and honestly, without fabrication or manipulation of results. Figure 2 displays a sample of the data.

Out[9]:

	Destination Port	Flow Duration	Total Fwd Packets	Total Backward Packets	Total Length of Fwd Packets	Total Length of Bwd Packets	Fwd Packet Length Max	Fwd Packet Length Min	Fwd Packet Length Mean	Fwd Packet Length Std	...	min_seg_size_forward	Active Mean	Active Std	Active Max	Active Min	Idle Mean	Idle Std	Idle Max	Idle Min	Label
0	54865	3	2	0	12	0	6	6	6.0	0.0	...	20	0.0	0.0	0	0	0.0	0.0	0	0	BENIGN
1	55054	109	1	1	6	6	6	6	6.0	0.0	...	20	0.0	0.0	0	0	0.0	0.0	0	0	BENIGN
2	55055	52	1	1	6	6	6	6	6.0	0.0	...	20	0.0	0.0	0	0	0.0	0.0	0	0	BENIGN
3	46236	34	1	1	6	6	6	6	6.0	0.0	...	20	0.0	0.0	0	0	0.0	0.0	0	0	BENIGN
4	54863	3	2	0	12	0	6	6	6.0	0.0	...	20	0.0	0.0	0	0	0.0	0.0	0	0	BENIGN

5 rows × 79 columns

Figure 2: Sample of CICIDS2017 Data

Data Preprocessing and Feature Engineering

Standard preprocessing and feature engineering were applied to the dataset. The pipeline involved cleaning column names, binarizing the target label (Benign=0, Attacks=1), imputing missing/infinite values using the median, and dropping non-predictive identifiers. Categorical variables were one-hot encoded, and numerical features were standardized via StandardScaler. To reduce dimensionality and improve computational efficiency, Random Forest feature importance was used to select the top-ranked predictors. Finally, to enable the LSTM to capture temporal dependencies, the tabular data was transformed into sliding-window sequences, with the final observation's label serving as the target.

Model Development Environment

The implementation was carried out in a Python-based analytical environment using the Anaconda distribution and Jupyter Notebook. Python was selected because of its extensive support for data analysis, ML, DL, and visualization. Several Python libraries were employed: Pandas was used for data loading, cleaning, and transformation, while NumPy supported numerical computation. Scikit-learn was used for data splitting, feature scaling, RF classification, and performance evaluation. TensorFlow and Keras were used to build and train the LSTM network. Statistical testing was performed using functions from SciPy, while Matplotlib was used for visualizing model behavior and results. Joblib was used to save trained models and preprocessing objects for future reuse. The use of this environment ensured reproducibility and allowed the development of a complete end-to-end pipeline for anomaly detection. The environment also supported experimentation with different model configurations, evaluation metrics, and statistical tests, which are essential for validating the proposed framework.

Model Development Algorithms

The ML algorithms developed and implemented for network anomaly detection in this study include: Random Forest (RF), Long Short-Term Memory (LSTM), and a hybrid model (RF-LSTM). The proposed model development process follows a hybrid learning strategy that combines RF and LSTM in an adaptive framework. The overall workflow includes: preprocessing the data, engineering the features, training RF, training LSTM on sequential windows, combining predictions through weighted fusion, updating the fusion strategy adaptively, and evaluating the final output using classification and statistical metrics.

The first stage involves training a RF classifier on the selected features. RF is used not only as a predictive model but also as a feature ranking mechanism. Its ensemble structure makes it effective for handling high-dimensional data and identifying the most relevant variables for classification. The probability output of the trained RF model represents its confidence that a given traffic instance is anomalous. The second stage involves constructing and training an LSTM model. Since LSTM is designed for sequential learning, the preprocessed tabular data is reorganized into overlapping time windows. Each window contains a sequence of flow records, allowing the model to learn temporal relationships among consecutive observations and output a probability score indicating the likelihood that the input sequence corresponds to malicious activity.

The third stage is hybrid fusion. The outputs of the RF and LSTM models are combined using a weighted averaging strategy. Let ($P_{\{RF\}}$) denote the probability produced by RF, and ($P_{\{LSTM\}}$) denote the probability produced by LSTM. The hybrid probability is computed as:

$P_{\{Hybrid\}} = \alpha P_{\{RF\}} + (1 - \alpha) P_{\{LSTM\}}$ where (\alpha) is a weighting parameter between 0 and 1 that controls the relative influence of the two models. A classification decision is then made by applying a threshold to the hybrid probability, typically 0.5. To incorporate adaptability, the framework adjusts the fusion weight dynamically based on recent model performance. This adaptive mechanism is intended to improve responsiveness to concept drift and changing traffic distributions. A sliding-window approach is used to evaluate recent predictions and adjust the contribution of each model over time. This is particularly important in cloud environments where network behavior evolves continuously.

Model Hyperparameter Configuration

In this study, hyperparameters were selected based on empirical testing, model stability, and alignment with the computational constraints of the dataset. For the RF model, the number of trees was set to 200, providing a balance between classification performance and computational cost. A larger number of trees generally improves robustness, but excessively large ensembles may slow training and inference without substantial gains. The random state was fixed to ensure reproducibility, and parallel processing was enabled to improve training efficiency.

For the LSTM model, two stacked LSTM layers were used. The first layer contained 64 units and returned full sequences to allow the second LSTM layer to process temporal

representations. The second LSTM layer contained 32 units and produced the final hidden state. Dropout layers with a rate of 0.2 were inserted after each LSTM layer to reduce overfitting. The output layer used a sigmoid activation function because the problem is a binary classification task. The model was compiled using the Adam optimizer, which is widely used for DL because of its adaptive learning rate and stable convergence behavior. Binary cross-entropy was chosen as the loss function because it is appropriate for binary classification problems. The number of epochs was set to 20, while the batch size was set to 64. Early stopping was also included to prevent overfitting by stopping training when validation loss no longer improved. The sliding window size used for sequence generation was set to 10; each LSTM input sequence consisted of 10 consecutive records. This window size was selected as a practical balance between capturing short-term temporal dependencies and avoiding excessive sequence length. The adaptive fusion weight (\alpha) was initially set to 0.5 to give equal importance to both models before dynamic adjustment, allowing the hybrid model to begin with balanced contributions from the RF and LSTM components.

Performance Metric

The proposed framework was evaluated using standard classification and statistical metrics to assess its predictive quality and adaptability. Accuracy (measures the proportion of correctly classified instances among all predictions), Precision (measures the proportion of predicted attacks that are actually attacks), Recall (measures the proportion of actual attacks that are successfully detected), F1-Score (harmonic mean of precision and recall; provides a balanced measure of classification performance), Confusion Matrix (class-wise prediction analysis) and ROC Curve and AUC (measures the model's ability to distinguish between benign and malicious classes across different classification thresholds; higher ROC-AUC indicates better discriminatory power).

Also, Statistical Significance Tests were used to determine whether differences in performance between models were statistically significant; the study used hypothesis testing procedures such as the Shapiro-Wilk test for normality, and the Wilcoxon signed-rank test for non-normal distributions.

RESULTS AND DISCUSSION

Focusing on evaluating the capability of the proposed framework to accurately distinguish benign traffic from anomalous traffic under dynamic network conditions, the findings obtained from the implementation and evaluation of the proposed system are presented here.

Dataset Distribution Analysis

Exploratory analysis was conducted to understand the characteristics of the dataset before model training. Figure 3 presents the class distribution of traffic instances within the dataset.

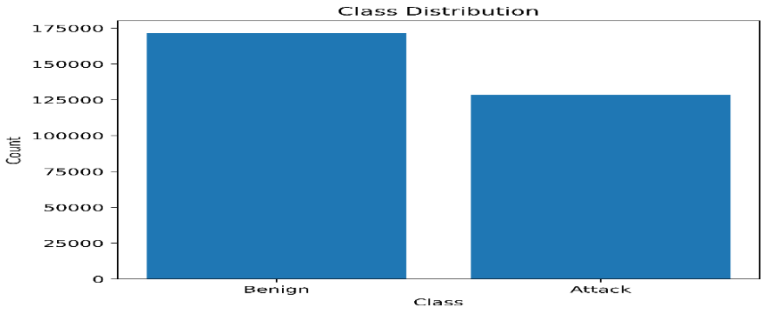

Figure 3: Class Distribution of Network Traffic

Figure 3 shows the class distribution of the CICIDS2017 dataset after binary label transformation, comprising Benign and Attack traffic. The benign class has approximately 172,000 instances, while the attack class has about 128,000 instances, indicating a moderate class imbalance; benign traffic is more prevalent, but not severely enough to bias learning significantly. The substantial number of attack samples ensures models are exposed to sufficient anomalous patterns during training. This distribution realistically reflects cloud environments where legitimate traffic typically exceeds malicious traffic, making the dataset suitable for developing robust anomaly detection models under practical conditions.

Feature Engineering and Importance Analysis

Feature engineering was performed to improve model efficiency and enhance predictive capability. RF feature importance analysis was used to identify the most informative network traffic attributes. Figure 4 presents the top-ranked features selected by the RF feature selection process.

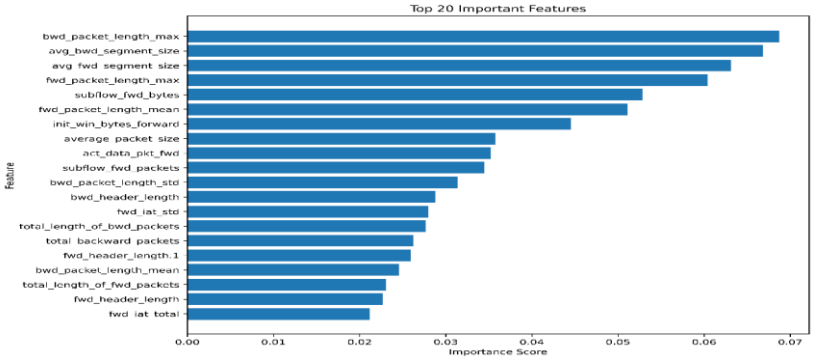

Figure 4: Top 20 Important Features

Figure 4 displays the top twenty most influential features from the RF importance analysis, where scores reflect each feature's contribution to classifying benign versus malicious traffic. The highest-ranked features, such as bwd_packet_length_max, avg_bwd_segment_size, avg_fwd_segment_size, and fwd_packet_length_max, along with others like subflow_fwd_bytes, fwd_packet_length_mean, average_packet_size, and init_win_bytes_forward, highlight that packet length characteristics, segment sizes, flow statistics, and transmission attributes are key indicators of anomalous behavior. The dominance of packet length and segment size metrics suggests that attack traffic follows different transmission patterns than normal traffic. Only a subset of all features substantially aids detection, justifying RF as a feature engineering tool within the hybrid framework. By selecting the most discriminative attributes, dimensionality is reduced, redundant information is removed, computational efficiency improves, and the LSTM model learns temporal patterns from informative features rather than the entire feature space.

Random Forest Model Performance

The RF classifier was first evaluated independently to establish a baseline performance benchmark for comparison with the proposed hybrid framework.

Table 1 presents the evaluation metrics obtained from the RF model.

Table 1: Random Forest Performance Metrics

Metric	Value
Accuracy	0.9998%
Precision	0.9999%
Recall	0.9998%
F1-Score	0.9998%
ROC-AUC	0.9998%

Table 1 shows that the RF classifier achieved outstanding performance on the CICIDS2017 dataset, with results indicating nearly perfect classification, very few false positives, and excellent discrimination between benign and malicious traffic. The close alignment of all metrics reflects a well-balanced classifier, validating RF as a strong baseline and effective feature selection mechanism within the proposed hybrid framework. Figure 5 visualizes the confusion matrix for the RF classifier.

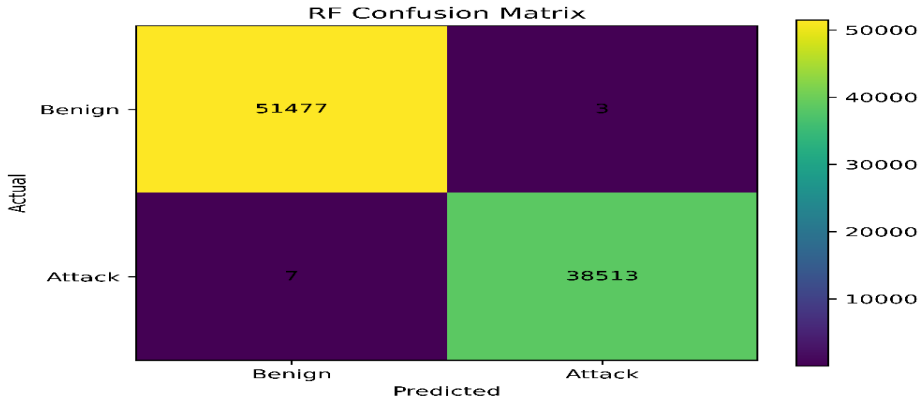

Figure 5: Random Forest Confusion Matrix

Figure 5 presents the confusion matrix for the RF classifier on the CICIDS2017 dataset. The model correctly classified 51,477 benign instances and 38,513 attack instances, with only 3 false positives (benign misclassified as attack) and 7 false negatives (attack missed). The dominance of values along the diagonal reflects near-perfect accuracy. The extremely low false positive rate is especially valuable in cloud environments, as it minimizes unnecessary alerts, reduces administrator workload, and prevents alert fatigue. Similarly, the low false-negative rate demonstrates strong attack detection capability, which is critical for maintaining cloud security by ensuring that few malicious activities go undetected. The confusion matrix confirms that RF achieves outstanding classification performance, validating its role as a core component of the proposed adaptive hybrid framework and providing a solid foundation for the LSTM model to further enhance detection through temporal dependency learning.

LSTM Model Performance

The LSTM model was trained using sequential traffic windows to capture temporal dependencies and evolving network behaviors. The learning curves obtained during training are presented in Figures 6 and 7.

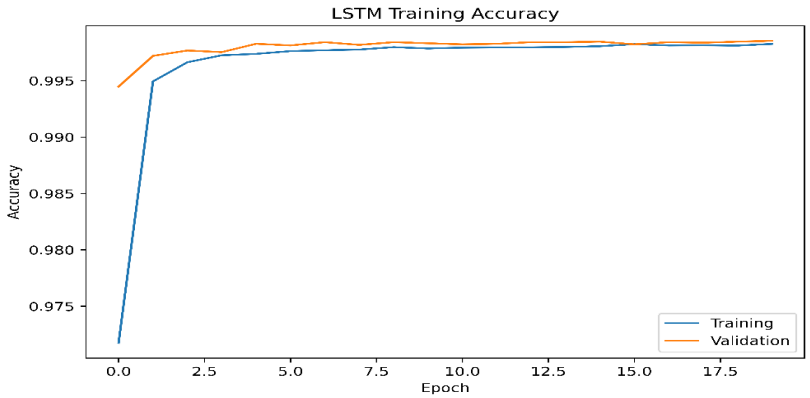

Figure 6: LSTM Training Accuracy Curve

Figure 6 shows the LSTM model's training and validation accuracy over 20 epochs. Accuracy rose rapidly from about 97.4% in the first epoch to over 99.5% by the second, eventually stabilizing at approximately 99.8%. The close alignment between training and validation curves, with no significant gap, indicates strong generalization and minimal overfitting. The stable validation accuracy after early epochs confirms that the LSTM reliably captured temporal dependencies in network traffic, supporting its integration into the proposed hybrid framework.

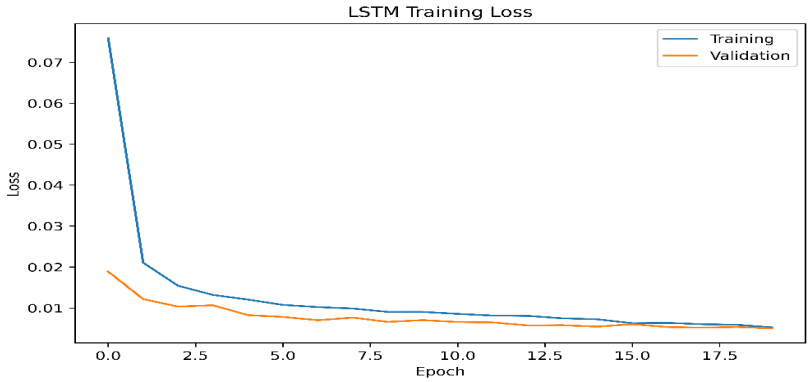

Figure 7: LSTM Training Loss Curve

Figure 7 presents the corresponding training and validation loss curves. Loss declined sharply from roughly 0.073 in the first epoch to below 0.02 within a few epochs, reflecting rapid minimization of prediction errors. Both loss curves gradually decreased to similarly low values, with validation loss closely tracking training loss. A small, temporary increase around epoch 16 was quickly resolved, indicating no significant overfitting. The convergence to low, stable loss values confirms that the LSTM learned robust, generalized representations of temporal attack patterns, making it well-suited for anomaly detection in dynamic cloud environments. The training and validation accuracy curves indicate stable learning behavior throughout the training process. Table 2 presents the evaluation metrics for the standalone LSTM model.

Table 2: LSTM Performance Metrics

Metric	Value
Accuracy	0.9984%
Precision	0.9990%
Recall	0.9973%
F1-Score	0.9982%
ROC-AUC	0.9983%

Table 2 shows that the LSTM model achieved strong performance on the CICIDS2017 dataset. The high precision minimizes false alarms, reducing unnecessary investigation costs in cloud environments. The recall of 99.73% indicates that nearly all actual attacks were detected, which is critical for preventing security breaches. The F1-score reflects an excellent balance between precision and recall, while the ROC-AUC confirms outstanding class separability across different thresholds. The results demonstrate that the LSTM effectively captures temporal dependencies in network traffic.

Hybrid RF–LSTM Framework Performance

The outputs of the RF and LSTM models were combined using probabilistic weighted fusion to form the hybrid RF–LSTM framework. Table 3 presents the performance metrics of the hybrid model as against the RF and LSTM models.

Table 3: RF, LSTM and Hybrid RF–LSTM Performance Metrics

Metric	RF	LSTM	Hybrid RF-LSTM
Accuracy	0.9998%	0.9984%	0.9996%
Precision	0.9999%	0.9990%	0.9996%
Recall	0.9998%	0.9973%	0.9994%
F1-Score	0.9998%	0.9982%	0.9995%
ROC-AUC	0.9998%	0.9983%	0.9996%

Table 3 shows that the proposed Hybrid Random Forest–LSTM model achieved outstanding performance on the CICIDS2017 dataset. The high precision minimizes false positives, reducing alert fatigue for security analysts, while the recall of 99.94% demonstrates excellent sensitivity in detecting nearly all actual attacks, critical for preventing security breaches. The F1-score confirms an exceptional balance between precision and recall, and the ROC-AUC reflects near-perfect class separability. Compared to the standalone LSTM model in Table 2, the hybrid framework shows consistent improvements across all metrics. These gains indicate that combining Random Forest's feature selection and nonlinear decision boundaries with LSTM's temporal dependency modeling enhances the detection of complex, evolving attacks. Table 3 provides strong empirical evidence that the Adaptive Hybrid RF–LSTM framework outperforms the LSTM baseline and offers a highly reliable solution for anomaly detection in dynamic cloud environments.

Adaptive Hybrid Framework Analysis

To improve adaptability in dynamic cloud environments, an adaptive weighting mechanism was introduced to dynamically adjust the contribution of RF and LSTM predictions. Figure 8 illustrates the adaptive alpha behavior across network flow sequences.

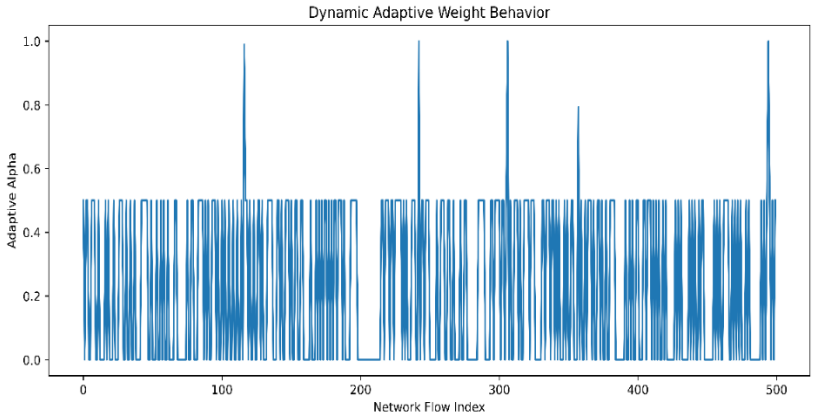

Figure 8: Adaptive Weight Dynamics

Figure 8 shows the dynamic behavior of the adaptive weighting parameter (α) across network flow instances in the proposed Adaptive Hybrid RF–LSTM framework. The weight fluctuates throughout the sequence, with most values alternating between 0 and 0.5 and occasional peaks approaching 1.0, suggesting a balanced influence from both models, allowing feature-based and temporal learning to complement each other. Sharp peaks close to 1.0 occur when one model is deemed more reliable for a particular traffic pattern, which is especially beneficial in dynamic cloud environments where traffic characteristics change rapidly. The absence of long constant-weight periods confirms the mechanism's responsiveness. Figure 8 validates that the adaptive weighting component dynamically balances model contributions, enhancing the robustness, flexibility, and detection performance of the hybrid framework. Figure 9 presents the anomaly prediction behavior over time.

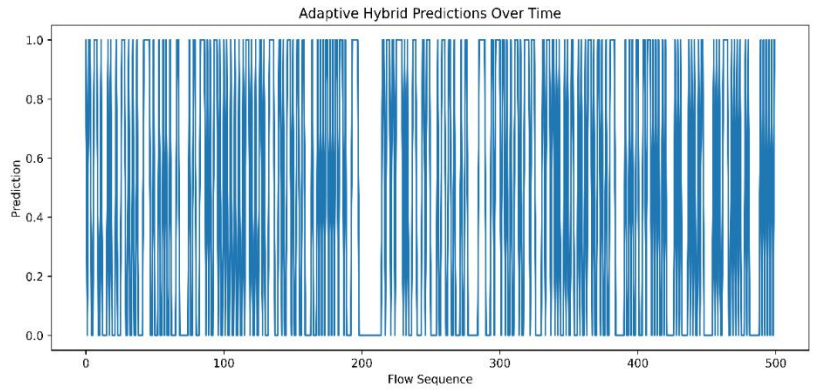

Figure 9: Adaptive Hybrid Predictions over Time

Figure 9 shows the prediction behavior of the Adaptive Hybrid RF–LSTM model across a sequence of network flows, where 0 represents normal traffic, and 1 represents anomalous traffic. The plot reveals frequent transitions between normal and anomalous classifications, indicating that the model continuously evaluates incoming traffic and dynamically responds to changes in network behavior. Predictions are clearly assigned to either 0 or 1 with no intermediate values, reflecting high confidence and supporting the strong performance metrics reported earlier. The balanced occurrence of normal and anomaly predictions demonstrates the model's ability to correctly recognize both benign and malicious patterns, while the distributed nature of anomaly detections across the entire sequence confirms robustness and lack of bias. The adaptive weighting mechanism influences these context-aware decisions, enabling consistent detection under varying conditions. Figure 9 validates that the Adaptive Hybrid RF–LSTM framework effectively monitors traffic in real time, making it well-suited for dynamic cloud environments.

ROC-AUC Analysis

Receiver Operating Characteristic (ROC) analysis was conducted to evaluate the discrimination capability of the models across varying classification thresholds. Figure 10 presents the ROC curves of the RF and Adaptive Hybrid models.

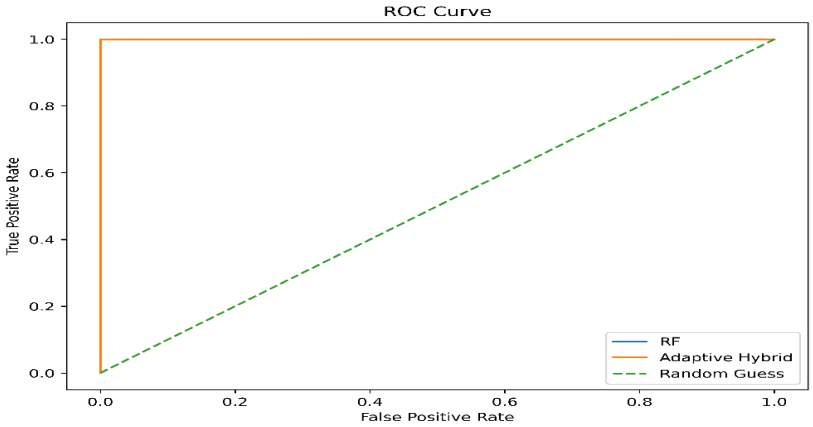

Figure 10: ROC Curve Comparison

The ROC curves of the suggested Adaptive Hybrid RF–LSTM framework and the baseline RF model are contrasted in Figure 10. Both curves show remarkably high discrimination between regular and anomalous traffic, closely following the upper-left corner of the graph, considerably above the diagonal random classifier line. By combining RF's feature selection with LSTM's temporal learning, the Adaptive Hybrid framework outperforms RF by a margin. The curves show an almost ideal balance between high detection rates and minimal false positives, with an AUC close to 1.0. The ROC analysis demonstrates that the adaptive hybrid framework outperforms conventional standalone models in real-time anomaly identification in dynamic cloud environments.

Statistical Significance Analysis

To determine whether the observed performance improvements were statistically significant, hypothesis testing was conducted using the Shapiro-Wilk normality test and Wilcoxon signed-rank test. Table 4 presents the statistical test results.

Table 4: Statistical Significance Test Results

Test	Statistic	P-value	Decision
Shapiro-Wilk	0.0043	3.12×10^{-181}	Non-normal
Wilcoxon	0.0000	7.24×10^{-8}	Significant

The performance data's non-normal distribution was verified using statistical tests. The Shapiro-Wilk test rejected normality and supported non-parametric approaches with a statistic of 0.0043 and a p-value of 3.12×10^{-111}, both of which were significantly below 0.05. A statistically significant difference between paired observations was indicated using the Wilcoxon signed-rank test, which produced a statistic of 0.0000 and a p-value of 7.24×10^{-2}. These results bolster the validity of the experimental findings by demonstrating that the performance improvements of the suggested Adaptive Hybrid Random Forest–LSTM Framework are significant and not the result of chance. Figure 11 visualizes a comparison of the baseline RF model and the proposed Adaptive Hybrid RF–LSTM framework.

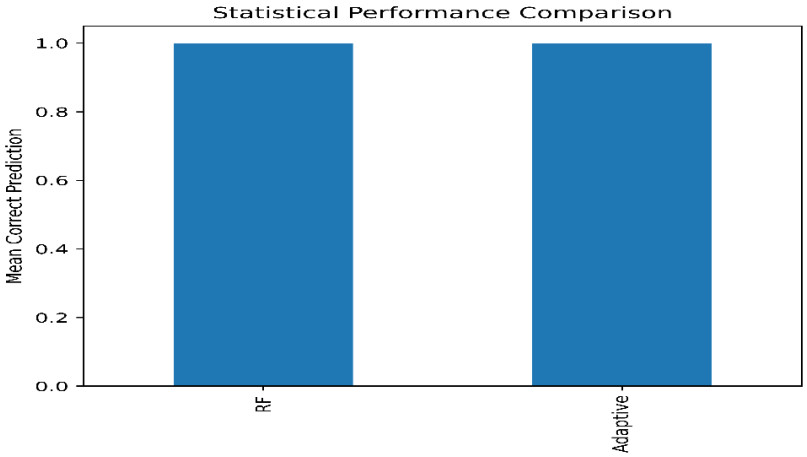

Figure 11: Statistical Performance Comparison

Figure 11 compares the mean correct prediction rates of the baseline RF model and the proposed Adaptive Hybrid RF–LSTM framework. Both models achieved near-perfect accuracy (close to 100%), indicating excellent classification on the CICIDS2017 dataset. Although the visual difference between the bars is negligible, the Wilcoxon signed-rank test ($p < 0.05$) confirmed a statistically significant difference between the two models' prediction outcomes. This demonstrates that even small performance variations are meaningful and not due to chance. The figure underscores the importance of complementing visual comparisons with statistical testing, especially when performance values cluster near the maximum. For dynamic cloud environments, the statistically significant improvement supports the adaptive hybrid framework's ability to combine RF's feature discrimination with LSTM's temporal learning, enhancing robustness and effectiveness in intrusion detection.

Discussion

This study developed and evaluated an Adaptive Hybrid Random Forest–LSTM Framework for network anomaly detection in dynamic cloud environments. The findings show that highly accurate, robust, and adaptive detection may be achieved by integrating temporal DL with feature-based ML and adaptive weighting. With 99.98% accuracy, precision, recall, F1-score, and ROC-AUC, standalone RF performed flawlessly. A confusion matrix revealed very few false positives and negatives. Although RF's dependability for high-dimensional intrusion detection was confirmed, adding LSTM was justified due to its inability to describe temporal dependencies. With 99.84% accuracy, 99.90% precision, 99.73% recall, 99.82% F1, and 99.83% ROC-AUC, the standalone LSTM showed steady learning, quick convergence, and little overfitting. Despite significantly worse raw performance, the LSTM was able to extract temporal dependencies and sequential attack patterns, which complemented RF's static feature analysis. Achieving 99.96% accuracy, 99.96% precision, 99.94% recall, 99.95% F1, and 99.96% ROC-AUC, the hybrid RF–LSTM performed better than the standalone LSTM. Combining discriminative features with temporal sequence modeling created a richer behavioral representation, detecting both known and novel anomalies and supporting hybrid ML-DL approaches to overcome individual model limitations.

The adaptive weighing technique is the most notable contribution. The system dynamically modified RF and LSTM contributions based on current performance, in contrast to static fusion. Flexibility and resilience for dynamic cloud environments were improved by weight analysis, which revealed constant fluctuations, shifting toward the more dependable model under shifting circumstances, and balancing when both contributed complementarily, showing an active response to concept drift and changing traffic patterns. With detections dispersed across the observation period, anomaly prediction analysis showed frequent, confident transitions between normal and anomalous classes, showing sustained responsiveness appropriate for real-time applications. Both RF and adaptive hybrid models produced curves that hugged the upper-left corner, according to ROC analysis, with the adaptive hybrid model outperforming RF by a small but significant margin. In order to prevent alarm fatigue in cloud security operations, an ideal trade-off between high true positive rates and low false positive rates is confirmed by the almost perfect ROC-AUC. The Wilcoxon signed-rank test found statistically significant differences between prediction outcomes, demonstrating that performance gains are not the result of random variation, and the Shapiro-Wilk test confirmed non-normal distribution, supporting non-parametric approaches. This enhances scientific rigor, particularly when measures go close to the theoretical maximum.

CONCLUSION

This study developed an Adaptive Hybrid Random Forest–LSTM Framework to improve network anomaly detection in dynamic cloud environments, addressing the limitations of traditional systems in handling evolving threats. The methodology comprised adaptive hybrid fusion, feature engineering, and preprocessing using the CICIDS2017 dataset. The results demonstrated that while LSTM captured temporal dependencies and RF successfully identified important traffic features, their hybrid integration, enhanced by an adaptive fusion mechanism, produced superior detection performance across metrics like accuracy, precision, recall, F1-score, and ROC-AUC. Significant improvements were confirmed by statistical tests. The framework provides a reliable solution for cloud security by utilizing the advantages of both models and adjusting to shifting traffic patterns.

This study highlights the benefits of integrating feature-based and temporal learning by providing a novel adaptive hybrid anomaly detection system with a dynamic prediction fusion mechanism. The LSTM component's high computing requirements, offline testing, binary (rather than multiclass) classification, evaluation on a single dataset, and a fixed sequence window size are some of its drawbacks. Adopting

hybrid ML-DL techniques for cloud security, putting adaptive mechanisms in place, continuously retraining models, and incorporating the framework into SIEM systems are among the recommendations. To improve generalizability and scalability, future research should investigate multiclass classification, real-time deployment, Attention-LSTM or Transformer-based models, federated learning, adaptive deep ensembles, and cross-dataset assessments.

REFERENCES

Abdulhammed, R., Faezipour, M., Abuzneid, A., & Alessa, A. (2022). Effective feature selection and classification for network intrusion detection using Random Forest. *Journal of Cybersecurity and Privacy, 2*(3), 512-534.

Alouffi, B., Hasnain, M., Alharbi, A., Alosaimi, W., Alyami, H., & Ayaz, M. (2021). A systematic literature review on cloud computing security: Threats and mitigation strategies. *IEEE Access, 9*, 57792–57807.

Eguavoen, V. O., Olanrewaju, B. S., & Okafor, C. N. (2025). A hybrid CNN-LSTM and AdaBoost model for classifying intrusion in IoT networks. *FUDMA Journal of Sciences, 9*(5), 204–212.

Imrana, Y., Xiang, Y., Ali, L., & Abdul-Rahim, Z. (2021). A bidirectional LSTM deep learning approach for intrusion detection in cloud network traffic. *IEEE Access, 9*, 70846–70861

Khan, M. A., Khan, J., Al-Yasiri, A., & Alomari, A. (2023). Computational complexity and interpretability trade-offs in deep learning for network security: A critical review. *Future Generation Computer Systems, 141*, 383-400

Kumar, S., Dwivedi, M., Kumar, M., & Gill, S. S. (2024). A comprehensive review of vulnerabilities and AI-enabled defense against DDoS attacks for securing cloud services. *Computer Science Review, 51*, Article 100604.

Lo, W. W., Layeghy, S., & Portmann, M. (2023). A systematic review of machine learning approaches for network intrusion detection: From static to dynamic models. *Computers & Security, 125*, 103015.

Mayuranathan, M., Saravanan, S. K., Muthusenthil, B., & Samydurai, A. (2022). An efficient optimal security system for intrusion detection in a cloud computing environment using hybrid deep learning technique. *Advances in Engineering Software, 173*, Article 103236.

Mishra, S., Jain, T., & Sitaram, D. (2022). *A hierarchical approach to conditional random fields for system anomaly detection*. arXiv

Modi, N. (2025). AI-powered intrusion detection using CNN-LSTM for cloud and edge networks: A hybrid deep learning approach. *Research Square*.

Sajid, M., Malik, K. R., Almogren, A., Malik, T. S., Khan, A. H., Tanveer, J., & Rehman, A. U. (2024). Enhancing intrusion detection: A hybrid machine and deep learning approach. *Journal of Cloud Computing, 13*(1), Article 123.

Tabrizchi, H., & Kuchaki-Rafsanjani, M. (2020). A survey on security challenges in cloud computing: Issues, threats, and solutions. T*he Journal of Supercomputing, 77*(8), 8761-8802

Tufail, A., Namoun, A., Sen, A. A. A., Kim, K. H., Alrehaili, A., & Ali, A. (2021). Moisture computing-based internet of vehicles (IoV) architecture for smart cities. *Sensors, 21*(11), Article 3783.