iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** VI **Month of publication:** June 2026

DOI: https://doi.org/10.22214/ijraset.2026.83364

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VI June 2026- Available at www.ijraset.com

A Hybrid AI-Driven Cybersecurity Framework For Real-Time Threat Detection Using Big Data Analytics

Gyanendra Kumar Gautam[1], Dr. Uday Pratap Singh[2]
[1]Research Scholar, [2]Assistant Professor, Department of Computer Science & Application,
Mind Power University, Bhimtal, Uttarakhand, India

Abstract: ***The growing advancements in cloud computing, IoT devices, and high-speed communication networks have created a complicated and more frequent environment for cyberattacks, posing significant risks to computer security. It has been discovered that the conventional intrusion detection techniques are insufficient to deal with sophisticated and zero-day attacks since they are not flexible enough to manage large amounts of real-time traffic.***

Using machine learning, deep learning, and big data analysis approaches, this study will concentrate on creating a hybrid AI-based security system. To improve attack categorisation, feature learning, and anomaly detection of known and new attacks, the proposed framework combines the Random Forest (RF), 1-Dimensional Convolutional Neural Networks (1D-CNN), and One-Class Support Vector Machine (OCSVM) methods. Scalable distributed stream processing and cybersecurity analytics will be made possible by the use of Apache Kafka and Apache Spark Streaming. To enable intelligent threat analysis, automatic threat containment, and threshold optimisation, weighted ensemble and adaptive mitigation algorithms are also incorporated. The proposed hybrid system can achieve 99.2% detection rate, 99.0% precision, 99.4% recall, and 99.1% F1-score with few false alarms, according to the experimental result using the CICIDS2017 dataset. With an average detection latency of 142 ms, a processing capacity of 52,000 packets per second, and a response time of less than 150 ms, runtime analysis shows that the framework is, in fact, scalable.

The proposed method enables the development of a scalable, flexible, and effective cybersecurity solution to identify threats and automatically neutralise them.

Keywords: Cyber Security, Intrusion Detection System (IDS), Hybrid Machine Learning, One–Dimensional Convolutional Neural Network (1D-CNN), Random Forest, OCSVM, Big Data Analytics, Real-time detection, Apache Kafka, Apache Spark.

I. INTRODUCTION

Cyber intrusions are becoming increasingly complex and frequent due to growing digital infrastructures such as cloud computing, Internet of Things (IoT), and high-speed networking systems. There are two examples of modern attacks advanced persistent threats (APTs) and zero-day attacks that can process huge amount of data and analyse events instantly [1], [2]. Attacks without a signature or attempts by an attacker to alter a known malware signature are beyond the capabilities of intrusion detection systems (IDS) that rely on signatures to identify malicious activity.

ML and DL algorithms are employed in cybersecurity to identify anomalies, extract features, and categorise assaults in order to overcome the aforementioned restrictions [3], [5]. Several classifiers can be used in ensemble learning to improve the system's accuracy and resilience [6], [8]. Real-time security monitoring and large-scale threat analysis are now possible because of new developments in distributed streaming and big data analytics [11] - [13]. Unfortunately, rather than incorporating the concept of scalability along with adaptive mitigation approaches into a cohesive system, the majority of current research efforts tend to focus on improving detection efficiency [10], [14].

The suggested architecture fills the need by merging technologies such as Apache Kafka, Apache Spark Streaming, Random Forest, 1D-CNN, and OCSVM into an adaptive hybrid cybersecurity platform. Real-time streaming analysis, threat identification, adaptive analysis, and even cyber threat mitigation will all be possible with this system. The creation of an adaptive cybersecurity framework can be regarded as the research's primary contribution to this sector.

©IJRASET: All Rights are Reserved

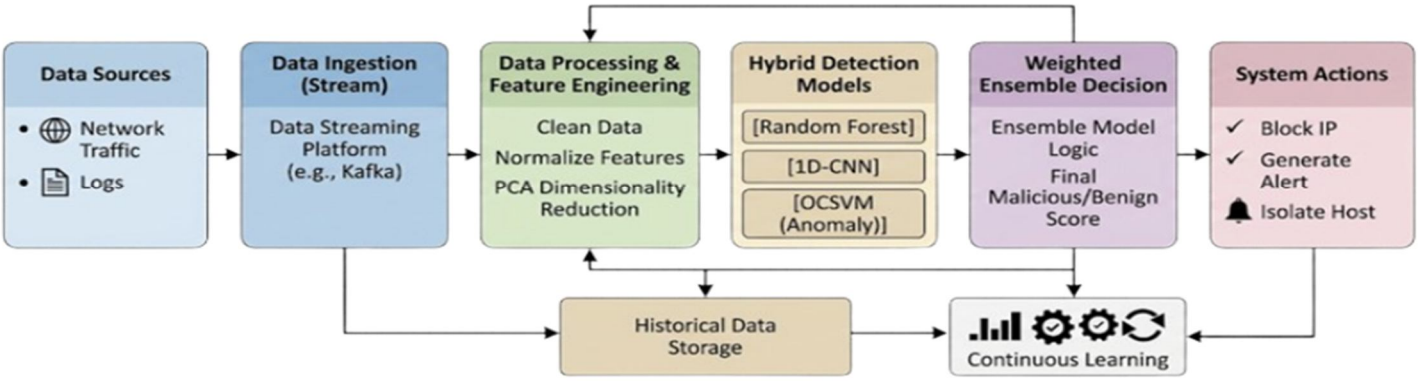


Fig. 1. Architecture of the proposed Hybrid Cybersecurity Framework

II. RELATED WORK AND RESEARCH GAP

Recent developments in cybersecurity technology have led to a rise in the use of machine learning techniques like deep learning (DL) for network intrusion detection. Traditional machine learning methods like Random Forest, Decision Trees, and SVM have demonstrated efficacy in identifying known attacks. Nevertheless, these techniques are not appropriate for identifying zero-day attacks because they need tagged datasets. Anomaly-based methods like OCSVM and Isolation Forest have been developed to solve this issue. These techniques are effective at identifying unidentified attacks. Unfortunately, system reliability is impacted by this method's high false positive rates [3], [6].

Deep learning-based approaches such as CNN and RNN are quite capable of capturing complex features in network traffic data, both in terms of space and time. CNN models are capable of automatically identifying malicious attack patterns and extracting characteristics from network traffic. Hybrid AI-based methods that combine machine learning, deep learning, and anomaly detection techniques offer improved detection capabilities, dependability, and attack adaptability by utilising a variety of algorithms [4], [7], [8]. Additionally, IDSs have been made more transparent through the application of Explainable Artificial Intelligence (XAI) approaches [9], [10]. Big data technologies like Apache Kafka and Apache Spark have become more relevant in terms of real-time distributed and scalable processing due to the increase in network traffic [11]–[13]. Even if all of these are happening, the majority of the research does not provide a framework that integrates the ideas of automated adaptation, intelligent detection using hybrid AI techniques, and real-time streaming analysis. To achieve real-time threat detection, adaptive mitigation, scalability, and enhanced zero-day attack detection in dynamic environments, the work presented fills a gap in the literature by presenting an adaptive security framework that includes techniques like Random Forest, 1D-CNN, OCSVM, Apache Kafka, and Apache Spark Streaming [16], [17].

Table 1. Analysis between Existing Methods and the Hybrid Framework

Method	Technique Type	Strengths	Limitations
Random Forest [1]	Supervised Learning	High Accuracy, Robust to noise	Requires labelled data
Support Vector Machine [2]	Supervised Learning	Effective in high dimensions	Poor performance on large datasets
OCSVM [3]	Unsupervised	Detects unknown attacks	High false positive rate
Isolation Forest [6]	Unsupervised	Efficient anomaly detection	Sensitive to parameter tuning
Suggested Hybrid Framework	Hybrid	Detects both known and unknown attacks, improves accuracy, reduces false positives, and enables adaptive decision-making.	Increased computational complexity and parameter tuning requirements.

Table 1 shows that current methods are not sufficient in terms of scalability, real-time capability, and zero-day threats. The suggested hybrid structure validates scalable and real-time cybersecurity resolution by integrating stream computing with intelligent learning-based models to address these issues.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VI June 2026- Available at www.ijraset.com

III. PROPOSED HYBRID DETECTION FRAMEWORK AND METHODOLOGY

This suggested model is a multi-layered, real-time cyber defence system that includes automated mitigation components, feature engineering, hybrid machine learning, and big data streams. It offers adaptive detection for both known and undiscovered security risks and is built to manage incredibly high network traffic volumes.

The system's continuous feedback loop, which allows it to adjust and alter behavior when a new class of risk vectors is discovered, must be the most crucial part of the architecture.

A. Framework Overview

The suggested cybersecurity framework is a multi-level, scalable design that will help identify and counteract cybersecurity risks arising from various network infrastructures. Real-time data collection, real-time big data stream processing, feature engineering, hybrid threat detection, adaptive decision-making, and threat response are all part of this cybersecurity framework. When it comes to quickly and reliably consuming real-time data from streaming network traffic and system logs, Apache Kafka is crucial. In the meantime, real-time cybersecurity dataset analysis is made possible by Apache Spark Streams [1], [2]. Recent research has shown that stream analytics can assist in enhancing performance for intrusion detection in heavily frequented networks [16]. Moreover, both known and new cyberthreats have been successfully identified by hybrid artificial intelligence cybersecurity systems that combine machine learning algorithms with anomaly detection [17]. The feature engineering module enhances data quality by implementing pre-processing techniques like missing value management, normalisation, feature selection, and dimensionality reduction. The following instance expose traffic statistics following pre-processing:

$$X = \{x_1, x_2, x_3, \ldots, x_n\}$$

Here, X denotes the feature vector and $x_1, x_2, x_3, \ldots, x_n$ are the features of network traffic that are taken into consideration for identifying threats utilizing the hybrid machine learning approach [3]. The CICIDS2017 data set, which is used to train and test the model, contains a substantial amount of data that is reflective of actual network activity.

B. Hybrid Threat Detection Model

The new hybrid detection method effectively detects threats in large-scale computer networks by utilizing Random Forest (RF), One-Class Support Vector Machine (OCSVM), and One-Dimensional Convolutional Neural Network (1D-CNN). The Random Forest (RF) method is utilized in the supervised learning approach to classify attacks. 1D-CNN facilitates the automatic extraction of deep features. Because OCSVM detects abnormalities during unsupervised learning, it can identify undiscovered and zero-day threats. In order to create a system that is extremely accurate, scalable, adaptive, robust, and generalizable, all three approaches—supervised learning, deep learning, and anomaly detection are combined.

1) Random Forest Classification Model

Random Forest is the main algorithm technique utilized, which finds any known attack pattern. Using many decision trees and majority voting, this classifier operates. The Random Forest classifier prediction equation is:

$$P_{RF}(X) = \frac{1}{T}\sum_{i=1}^{T} h_i(X)$$

Here, $P_{RF}(X)$ is the random forest score. T is the total number of decision trees. $h_i(X)$ indicates the output of the i^{th} decision tree.
In network traffic, the random forest model shows excellent noise resistance and classification accuracy.

2) Deep Learning Model with 1D-CNN

The One-Dimensional Convolutional Neural Network (1D-CNN) identifies complex traffic patterns and extracts features. The expression for the CNN prediction function is:

$$P_{CNN}(X) = f(W * X + b)$$

Where, $P_{CNN}(X)$ denotes CNN Prediction Score, W indicates Convolution Weight, b represents Bias Value, $f()$ represents the Activation Function and * is the operator for Convolution.
The suggested CNN model can identify sophisticated attacks and learn characteristics more effectively.

©IJRASET: All Rights are Reserved

3) Anomaly Detection Based on OCSVM

The One Class Support Vector Machine (OCSVM) is employed to identify any departure from typical traffic behaviour. Unlike supervised techniques, OCSVM can successfully identify unknown and zero-day attacks.

The approach for OCSVM anomaly detection is:

$$P_{OCSVM}(X) = \sum_{i=1}^{N} \alpha_i K(X_i, X) - \rho$$

Where, $P_{OCSVM}(X)$ is the anomaly score and α_i is the support vector coefficients, $K(X_i, X)$ indicates Kernel function, ρ Offset of the decision border, N represents the quantity of support vectors.

The probability that network anomalies are malicious increases with the anomaly score. However, the likelihood of typical network traffic behaviour increases with a lower anomaly score.

C. Ensemble Weighted Decision Model

The output from CNN, RF, and OCSVM models is employed in a weighted ensemble method to generate output. The analysis of the ensemble score is as follows:

$$S(X) = w_1 P_{RF}(X) + w_2 P_{CNN}(X) + w_3 P_{OCSVM}(X)$$

such that: $w_1 + w_2 + w_3 = 1$

Where S(X) is the ensemble classifier's overall threat score. w_1, w_2, w_3 weights are allocated to the corresponding models according to validation scores. The outputs of separate models are $P_{RF}(X)$, $P_{CNN}(X)$, and $P_{OCSVM}(X)$. By using this technique, classification errors are reduced, and detection precision is increased.

D. The Classification Decision Rule and Threshold Initialisation

To identify unauthorized access activity, an adaptive threshold initialization technique is used. The decision rule and threshold initialization process are as follows:

1) Initialisation of Threshold

The CICIDS2017 database's normal and anomalous data samples are used to calculate the threshold value θ during the validation phase in order to provide a dependable and consistent decision-making process. Optimising the F1-Score while concurrently lowering the False Alarm Rate determines the threshold's initial value. The value of the first threshold is set to:

$$\theta_0 = 0.60$$

where the initial threshold value is represented by θ_0, and 0.60 is selected by an experimental validation procedure. At the initial execution stage, the threshold selected provides an ideal trade-off between false positives and attack detection performance.

2) Rule for Classification Decision Making

The first classification decision rule defined as:

$$Y(X) = \{1, if\ S(X) \geq \theta_0$$
$$0, otherwise\}$$

When Y(X) = 0 is normal traffic, S(X) is the ensemble threat score, and Y(X) = 1 represent malicious traffic. With this initiation mechanism, the framework can accurately classify threats in real-time earlier than adaptive thresholding.

E. Adaptive Mitigation Decision Model

The ensemble threat score S(X) will be utilized in a confidence-based adaptive mitigation technique to determine the appropriate course of action in response to threats. Depending on the degree of threat detected, the system automatically implements mitigation actions such as traffic monitoring, alert creation, IP address blocking, and node isolation.

1) Mitigation Decision Function

The mitigation action *M(X)* is specified as a piecewise function:

$$M(X) = \begin{cases} Block\ IP\ /Isolate\ Node, & if\ S(X) \geq \theta_1 \\ Generate\ Alert, & if\ \theta_2 \leq S(X) < \theta_1 \\ Monitor\ Activity, & if\ S(X) < \theta_2 \end{cases}$$

When θ_1 is the high-risk threshold, θ_2 is the medium-risk threshold, and S(X) is the hybrid detector's threat score. While less dangerous behaviours will be monitored, high-confidence threats will elicit quick action to reduce the risk.

2) Adaptive Threshold Update Mechanism

To improve adaptability in dynamic cybersecurity situations, the decision criteria are regularly changed using a feedback-based learning technique. The adaptive threshold update function has the following expression:

$$\boldsymbol{\theta_t = \theta_{t-1} + \alpha(E_t - E_{reference})}$$

where θ_t is the threshold's current value, θ_{t-1} is its previous value, α is the learning rate, E_t is the system error at this point, and $E_{reference}$ is the desired error level. Adaptive optimisation makes it easier to change decision boundaries in response to emerging cyber threats and system performance.

3) Automated Response and Mitigation

When a threat is detected, it is automatically mitigated using a set of techniques determined by its severity and level of confidence in order to maintain services and reduce system harm. Sandboxing potentially dangerous processes, limiting traffic to prevent DDoS attacks, quarantining infected nodes to reduce the spread of threats laterally throughout the system, and dynamic firewall adjustments to block malicious IPs are a few of these tactics. An adaptive alerting strategy is another component of the threat mitigation structure, whereby lower level attacks are continuously watched while higher priority warnings are raised for extremely dangerous attacks. Feedback mechanisms are used to continuously optimize these mitigation techniques.

F. Characteristics of the Framework

The proposed architecture performs real-time cybersecurity analysis by utilising big data technology and hybrid machine learning methods. The detection of both known and unknown cyberthreats is improved by the use of Random Forest, 1D-CNN, and OCSVM. Additionally, distributed processing systems may effectively handle massive volumes of network traffic by utilising Apache Kafka and Apache Spark Streaming.

IV. RESULTS AND DISCUSSION

The CICIDS2017 Dataset is used for experimentation, providing a benchmark with a realistic and widely recognised intrusion detection dataset in the scientific community. This dataset is a great starting point for testing new intrusion detection systems because it mimics both benign and malicious network activity. The CICFlowMeter is used to extract features from millions of network flow entries in the dataset, making it possible to successfully analyse the high-dimensional data in traffic flows. DDoS, brute force (FTP and SSH), botnet, infiltration, and web-based attacks are among the attack types that enable the assessment of zero-day attacks. Stratified sampling is used to split the modelling data into 80% training and 20% testing.

A. Data Processing

To provide a realistic simulation of network traffic, CICIDS2017 [15] has been chosen for this purpose. A suitable data preparation pipeline is required to increase the quality of the data used in this investigation. The following data pre-processing tasks are included in this process:

- Data Cleaning: In an effort to maintain the data in an ideal condition, any data that contains missing, null, or infinite values has been removed from the data set. Additionally, redundant and duplicate data have been eliminated.
- Normalisation: Min-Max normalisation was used to normalise all numerical attributes to a range of [0,1], ensuring balanced model training [2].
- Label Encoding: Since the categorical label data represent different classes of network traffic, it has been encoded into numerical forms.
- Feature Selection: Statistics and correlations have been used to remove features that are superfluous or less informative.

B. Experimental Setup

The proposed real-time cybersecurity solution is based on a distributed big data framework that makes it easier to analyse high-bandwidth traffic. The messaging queue software that enables real-time data intake is Apache Kafka [13]. The suggested approach uses Apache Spark Streaming as its processing layer to enable scalable parallel processing for massive data stream volumes [12].
Python-based tools like Scikit-Learn and Tensorflow are used to build machine learning and deep learning models, respectively. The suggested model will be experimentally evaluated on a system with a multi-core CPU that has sufficient computational memory. The hybrid technique is more complex, but distributed processing makes execution more efficient, enabling near real-time performance that can satisfy the demands of fast networks.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

C. Metrics for Evaluation

The proposed intrusion detection methodology was evaluated using classification metrics generated from the confusion matrix [2]. Among these metrics are:

- True Positive (TP): The quantity of attacks that the system correctly classifies.
- True Negative (TN): The quantity of typical traffic that the system accurately classified
- False Positive (FP): When regular traffic is mistakenly classified as an attack.
- False Negative (FN): Attack traffic that is mistakenly classified as regular traffic

1) Accuracy: The accuracy measure shows how well both normal and attack traffic are classified overall.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

2) Precision: The percentage of attacks that are accurately classified out of all those that are categorised as attacks is known as precision.

$$\text{Precision} = \frac{TP}{TP + FP}$$

3) Recall/Detection rate: This parameter determines how effectively the system can accurately classify attacks.

$$\text{Recall} = \frac{TP}{TP + FN}$$

4) F1 Score: It aids in striking a healthy balance between recall and precision, which is especially helpful when working with unbalanced data sets.

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

5) False Alarm Rate (FAR): FAR measures the proportion of lawful traffic that is mistakenly identified as harmful.

$$\text{FAR} = \frac{FP}{FP + TN}$$

6) Detection Rate (DR): Recall and DR are interchangeable.

$$\text{DR} = \frac{TP}{TP + FN}$$

D. Performance Evaluation

The proposed hybrid model is analysed using the CICIDS2017 database to determine its effectiveness in detecting cyber threats, which can be classed as known or unknown. The accuracy, precision, recall, and F1-score metrics are used to evaluate performance. Table 2 presents the findings of the performance comparison between the various models. The Random Forest model performs well due to the strength of ensemble learning, as evidenced by its accuracy score of 96.7%. Because of its ability to take highly complex elements in the network traffic data, the CNN model has an accuracy score of 97.8%. Despite that, the proposed hybrid model outperforms both models, with an accuracy score of 99.2%, a precision rate of 99.0%, a recall value of 99.4%, and an F1 score of 99.1%. These changes demonstrate the effectiveness of combining multiple learning methodologies into an integrated strategy.

Table 2. Comparative Evaluation of ML Models

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Random Forest	96.7	96.1	95.8	95.9
CNN	97.8	97.3	97.0	97.1
Hybrid Model	99.2	99.0	99.4	99.1

The merging of Random Forest, CNN, and OCSVM methods is responsible to improve performance. While CNN assists in getting deep learning features and Random Forest helps achieve high classification capabilities, OCSVM assists in detecting anomalies due to departures from typical traffic. The system's performance has been enhanced by the hybrid approach for both known and unknown assaults. Five-fold cross-validation was used to verify our system's performance and validate the experiment's findings. The average accuracy of our hybrid system is 99.2%, with a standard variation of ±0.3.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

E. Confusion Matrix Analysis

The confusion matrix provides a thorough evaluation of the classification accuracy of the proposed hybrid model. As shown in Table 3.

Table 3. Confusion Matrix of Proposed Model

	Predicted Attack	Predicted Normal
Actual Attack	TP = 9940	FN = 60
Actual Normal	FP = 100	TN = 9900

The huge amount of TP (9940) and TN (9900) demonstrates that the system correctly distinguishes between malicious and non-malicious traffic. In order to maintain system security, the low level of FP (100) lowers the quantity of false alarms. Additionally, the modest number of undiscovered attacks is shown by the low value of FN (60). For cybersecurity systems, this feature is essential because a false negative can have detrimental effects. Adding OCSVM to the framework is the main way to lower the number of FN. Such a solution enhances the identification of zero-day or unknown attacks because OCSVM is based on modelling typical behaviour and identifying deviations from it.

F. Analysing the ROC Curve for Comparative Performance

The ROC analysis aids in measuring the performance of these models by determining the relationship between the True Positive Rate (TPR) and the False Positive Rate (FPR). The ability of these models to distinguish between malicious and benign traffic is clarified by this investigation. The Area Under the Curve (AUC), which is high for models that perform well, is a significant metric that has been taken into consideration. The results are consistent with Table 2.

1) ROC Curve Analysis

The ROC curves for Random Forest, CNN, and the suggested Hybrid model are compared in Figure 2. Because its ROC curve is closest to the upper left corner, which indicates the highest TPR and lowest FPR, the suggested Hybrid model is outperforming other models in this regard. In this context, the suggested Hybrid model has the highest Area under the Curve (AUC) value of 0.99, while CNN and Random Forest models have AUC values of 0.98 and 0.97, respectively. The accuracy, precision, recall, and F1-score values shown in Table 2 are in line with these findings. The enhanced ROC curve shows that combining Random Forest, 1D-CNN, and OCSVM improves detection capabilities for both known and undiscovered cyberattacks while lowering false positives. Because OCSVM detects any departure from the typical behaviour of network traffic, its deployment in particular improves the accuracy of anomaly identification.

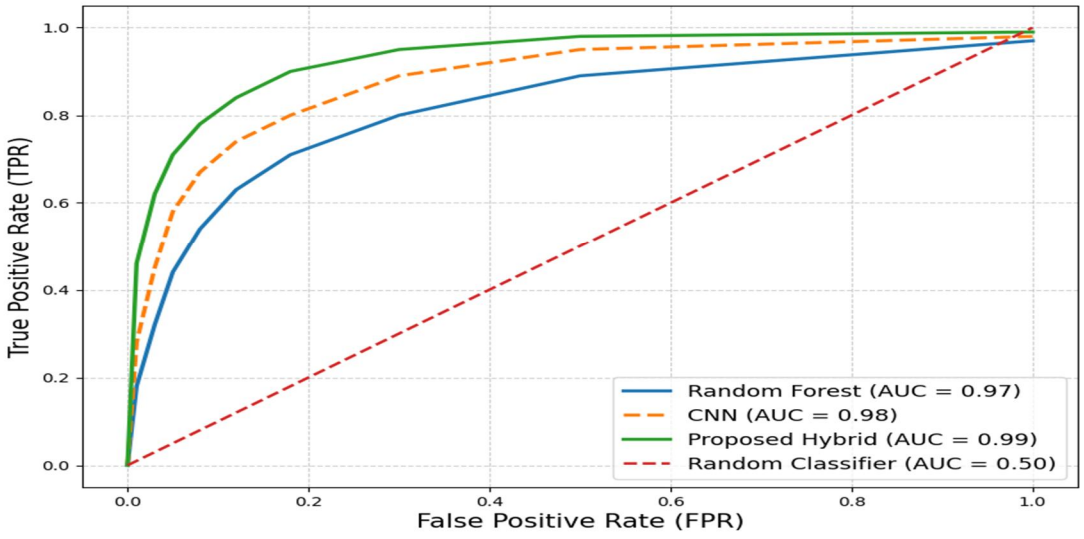


Fig 2. Comparison of Random Forest, CNN, and Hybrid Models' ROC curves

2) Analysis of Comparative Performance

A comparison with current deep learning and machine learning methods has been presented in figure 3, to further validate the efficacy of the proposed framework.

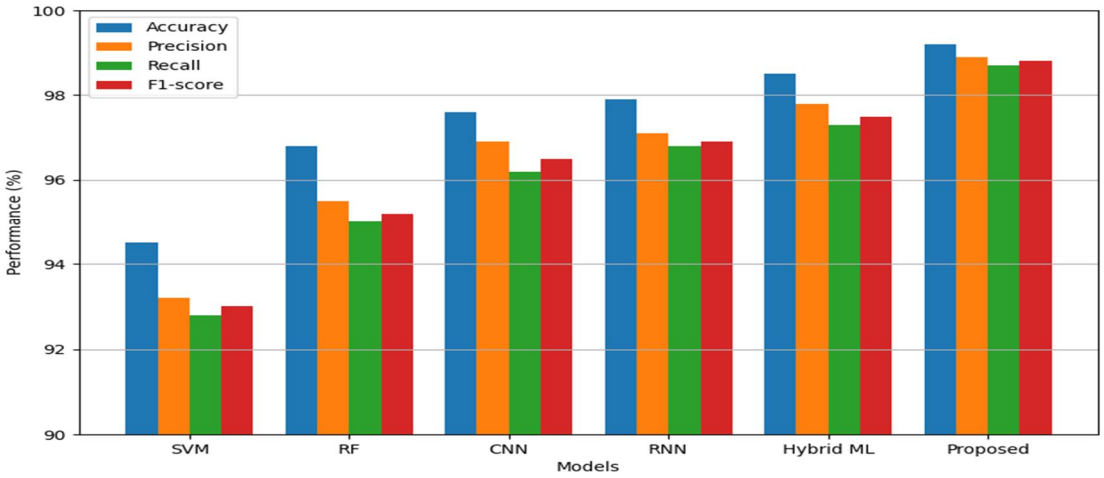


Fig 3. Performance analysis of the suggested hybrid model by component

The contributions of each hybrid method component are displayed in Figure 3. The findings clearly show that CNN does well in feature extraction and Random Forest does well in classification. OCSVM is more effective at identifying irregularities.

G. Analysis of Mitigation Effectiveness

The effectiveness of the proposed automated threat mitigation system was evaluated using a number of metrics, like response time, threat containment rate, and attack impact reduction. Attack sessions that were successfully mitigated were divided by the total number of attacks detected in order to get the threat containment rate. By examining the variations in malicious traffic propagation and network resource usage prior to and during mitigation implementation, the decrease in attack impact was quantified. The adaptive mitigation system illustrates real-time attack mitigation on high-speed networks by mitigating cyber threats within 150 milliseconds of their discovery, according to experimental studies. Additionally, a 96.5% decrease in the overall impact of cyberattacks demonstrates the efficacy of the suggested methodology.

H. Evaluation of Scalability and Runtime

To evaluate the usefulness and scalability of the recommended architecture, investigation was carried out using Apache Spark and Apache Kafka Streaming on huge network traffic from the CICIDS2017 dataset. Here, the suggested framework achieved a throughput rate of 52,000 packets per second and an average latency time of 142 milliseconds. The framework made use of Apache Spark Streaming for distributed parallel traffic processing and Apache Kafka for reliable real-time data intake. Because of the system's distributed stream-processing architecture, memory use remained constant while CPU usage did not surpass 78% during execution. The runtime and scalability evaluation of the recommended architecture is shown in Table 4.

Table 4: The runtime and scalability analysis of the proposed framework

Parameter	Observed value
Average Detection Latency	142 ms
Packet Throughput	52000 packets/sec
Kafka Stream Rate	185 MB/Sec
Spark Processing Delay	121 ms
Average CPU Utilisation	78%
Threat Response Time	<150 ms

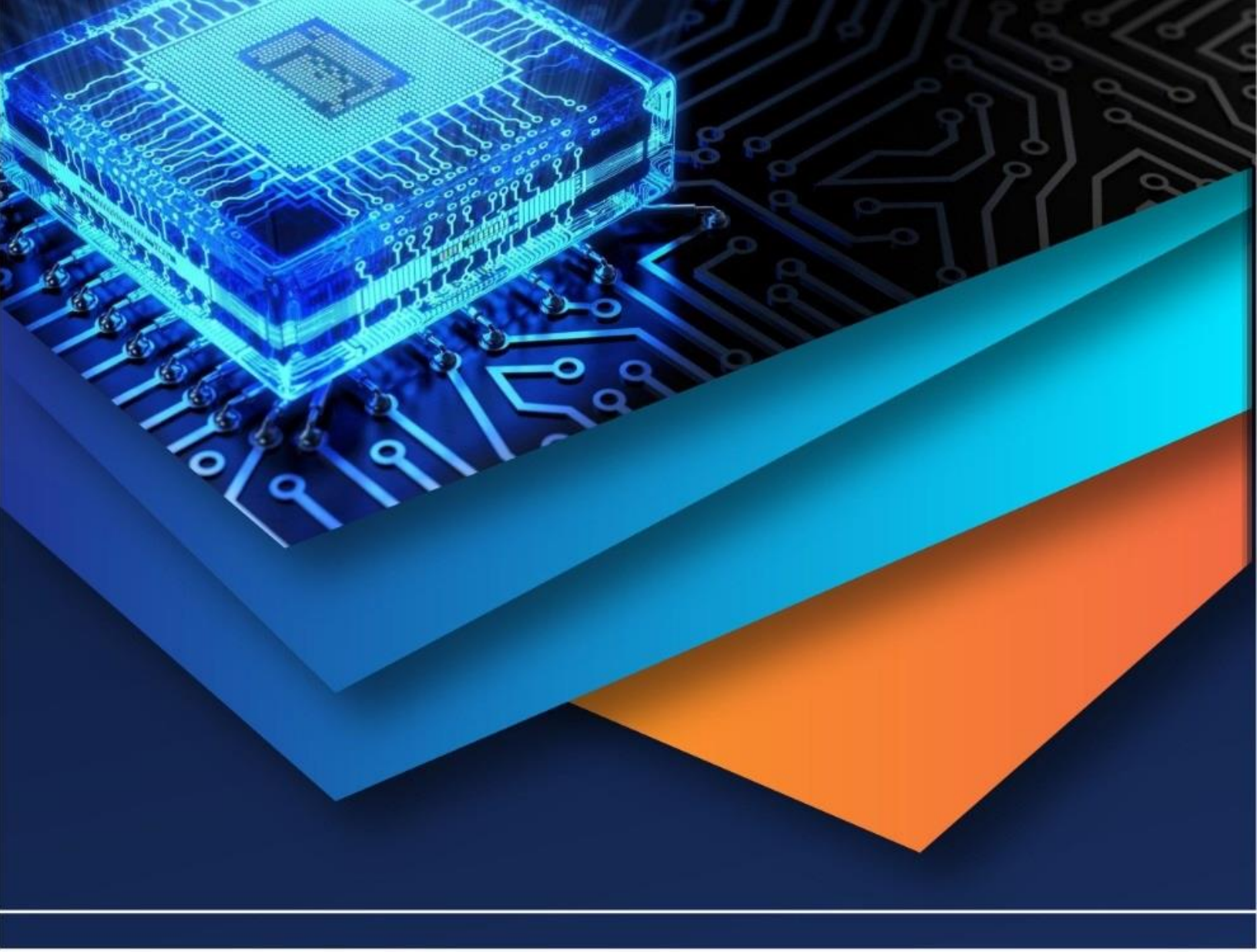
V. CONCLUSION AND FUTURE WORK

This study provides a refined cybersecurity architecture that makes use of big data, machine learning, deep learning, and anomaly detection. Especially, the suggested strategy connects streaming techniques and hybrid AI into a single framework that allows for real-time cybersecurity threat processing and analysis. This architecture aims to provide a flexible and scalable foundation for constructing more complicated solutions using modern AI technologies rather than a replacement for intrusion detection systems. The experimental conclusions show that the suggested method has excellent levels of accuracy, precision, recall, and F1-scores together with very little incorrect categorization. The suggested hybrid approach can improve the identification of both known and undiscovered cyberthreats. Moreover, by quickly containing the effects of an assault, one can guarantee better reaction to threats with the aid of the adaptive system response. However, despite all of the aforementioned benefits, the suggested strategy has certain drawbacks. First, the results reported in this study are derived from produced traffic rather than actual deployments in enterprise networks. Second, federated and limited environments are not supported by the design. Even though the suggested design has produced encouraging experimental results, it is necessary that it be put into practice in order to give additional performance tests when exposed to the dynamic cyber threats. To provide better interpretability and intelligence in detection, future research will emphasize on combining Explainable AI (XAI) with cutting-edge deep learning techniques. The system will also be combined with edge computing and Internet-of-things-based cybersecurity technologies to enable it to function effectively in decentralized and resource-constrained contexts.

REFERENCES

[1] S. A. Ajagbe, J. B. Awotunde, and H. Florez, “Intrusion detection: A comparison study of machine learning models using unbalanced dataset,” SN Computer Science, vol. 5, no. 6, Art. no. 1028, 2024, doi: 10.1007/s42979-024-03369-0.
[2] A. J. A. Immastephy, R. M. Noor, and M. A. Razzaque, “A systematic review on intrusion detection systems using machine learning,” E3S Web of Conferences, vol. 512, pp. 1–10, 2024.
[3] M. Benmalek, A. Kharrazi, and M. Mezghani, “Anomaly-based intrusion detection using unsupervised learning techniques,” Procedia Computer Science, vol. 225, pp. 1500–1509, 2024.
[4] S. Elsayed, K. Mohamed, M. A. Madkour, and M. A. Madkour, “A Comparative Study of Using Deep Learning Algorithms in Network Intrusion Detection,” IEEE Access, vol. 12, pp. 58851–58870, 2024, doi:10.1109/ACCESS.2024.3389096.
[5] S. Elouardi, A. Motii, M. Jouhari, A. N. H. Amadou, and M. Hedabou, “A Survey on Hybrid-CNN and LLMs for Intrusion Detection Systems: Recent IoT Datasets,” IEEE Access, vol. 12, pp. 180009–180033, 2024, doi: 10.1109/ACCESS.2024.3506604.
[6] P. Waghmode, S. Patil, and R. Kulkarni, “Feature selection and hybrid machine learning models for intrusion detection,” Scientific Reports, vol. 15, 2025.
[7] U. Shahid, M. Z. Hussain, M. Z. Hasan, A. Haider, J. Ali, and J. Altaf, “Hybrid Intrusion Detection System for RPL IoT Networks Using Machine Learning and Deep Learning,” IEEE Access, vol. 12, pp. 113099–113112, 2024, doi: 10.1109/ACCESS.2024.3442529.
[8] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, “Ensemble-based intrusion detection systems using machine learning,” Computers & Security, vol. 132, 2024.
[9] A. Alabbadi, H. Alkahtani, and M. Alshammari, “Explainable AI-based intrusion detection for IoT environments,” Sensors, vol. 25, no. 3, pp. 847–862, 2025.
[10] Y. Wang et al., “Explainable and adaptive cybersecurity frameworks using deep learning,” IEEE Access, vol. 13, pp. 11234–11250, 2025.
[11] X. Liu et al., “Real-time intrusion detection using big data analytics and streaming frameworks,” IEEE Access, vol. 12, pp. 20245–20260, 2024.
[12] M. Zaharia et al., “Apache Spark: A unified engine for big data processing,” Communications of the ACM, vol. 66, no. 11, pp. 56–65, 2023, doi: 10.1145/3610228.
[13] J. Kreps, N. Narkhede, and J. Rao, “Kafka: A distributed streaming platform for real-time data pipelines,” IEEE Data Engineering Bulletin, vol. 46, no. 2, pp. 20–29, 2023.
[14] R. Chinnasamy, P. Ramasamy, and S. Karthikeyan, “Challenges in integrating machine learning with big data for cybersecurity,” ICT Express, vol. 11, no. 2, pp. 210–218, 2025.
[15] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization,” in Proc. Int. Conf. Information Systems Security and Privacy (ICISSP), 2018, pp. 108–116, doi: 10.5220/0006639801080116.
[16] M. Sajid, K. R. Malik, A. Almogren, T. S. Malik, A. H. Khan, and A. U. Rehman, “Enhancing intrusion detection: a hybrid machine and deep learning approach,” Journal of Cloud Computing, vol. 13, Art. no. 123, 2024.
[17] Y. Imrana, Y. Xiang, L. Ali, A. Noor, and K. Sarpong, “CNN-GRU-FF: a double-layer feature fusion-based network intrusion detection system using convolutional neural network and gated recurrent units,” Complex & Intelligent Systems, vol. 10, pp. 3353–3370, 2024, doi: 10.1007/s40747-023-01313-y.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)