Journal of Computer Science and Technology Studies
ISSN: 2709-104X
DOI: 10.32996/jcsts
Journal Homepage: www.al-kindipublisher.com/index.php/jcsts

JCSTS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

An Intelligent Machine Learning Framework for Cybersecurity Threat Detection and Monitoring

Sasi Kiran Malladi
Independent Researcher, *Kakatiya Institute of Technology and Science*
Corresponding Author: Sasi Kiran Malladi, **E-mail**: skmalladi.tech@gmail.com

| ABSTRACT
As the number of IoT devices increases rapidly, cybersecurity threats are becoming more dynamic and intricate, necessitating more sophisticated and intelligent defense mechanisms. This work presents a unique intrusion detection system that enhances the security of the IoT by combining signature-based and anomaly-based techniques with AI-driven threat mitigation. The proposed framework is an amalgamation of Recurrent Neural Network (RNN) and Extreme Gradient Boosting (XGBoost) models that efficiently process network traffic, detect suspicious activities, and allow a real-time response to threats. According to the ToN-IoT dataset, data cleaning, normalization, and data balancing are employed to improve performance. The experimental analysis indicates the high-performance rate where the RNN and XGBoost have 98.5% and 98.7% accuracy respectively. The RNN model exhibits a marginally greater trade-off in terms of precision, recall and F1-score metrics. The efficacy of the proposed approach in the process of achieving high detection performance is validated by the comparative analysis with the currently existing models such as Naive Bayes, LSTM, CNN, and Random Forest. The research as a whole indicates the potential of DL and ensemble methods to improve the process of cybersecurity threat detection in the IoT context, as well as to address the issues of data imbalance and feature relevance.

| KEYWORDS

Cybersecurity, Threat Detection, Anomaly Detection, Data Monitoring, Risk Mitigation, Intrusion Detection, Machine Learning, Predictive Analytics

| ARTICLE INFORMATION

ACCEPTED: 09 April 2026 **PUBLISHED:** 23 May 2026 **DOI:** 10.32996/jcsts.2026.8.7.11

1. Introduction

In the modern globalized digital environment, the issue of cybersecurity has gained sharp significance for organizations, governments, and individuals. The explosion of digital infrastructure, IoT devices and cloud computing has dramatically expanded the size of data and the potential attack surface of cyber threats [1][2]. Conventional security systems, such as rule-based and signature-based Intrusion Detection Systems (IDS), are generally not sufficient to identify modern and advanced attacks[3]. The need to transition towards smarter and more adaptable security solutions is fueled by the fact that conventional systems are unable to identify zero-day vulnerabilities, polymorphic malware, and multi-stage cyber-attacks[4]. The dynamic character of cyber threats has given rise to a growing need for sophisticated methods that are capable of managing large-scale and complex data. Cybersecurity in the current world entails safeguarding networks, systems and sensitive information against unauthorized access and malicious activities[5][6]. Nevertheless, the high data dimensionality, dynamic patterns of attack, and real-time requirements of detection make this task challenging[7]. These restrictions highlight the need to create powerful frameworks that can automatically analyze data, identify anomalies, and respond to emerging threats [8]

Artificial Intelligence (AI) has become a radical solution to cybersecurity, allowing systems to go beyond fixed defenses into more intelligent and dynamic mechanisms[9]. Real-time data-driven decision-making, Anomaly Detection, and Pattern Recognition from past data are all possible with AI-based models[10]. ML and DL, as significant subsets of AI, are important to develop intelligent

Copyright: © 2026 the Author(s). This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) 4.0 license (https://creativecommons.org/licenses/by/4.0/). Published by Al-Kindi Centre for Research and Development, London, United Kingdom.

cybersecurity frameworks[11]. The supervised, unsupervised and Reinforcement Learning are all ML techniques that allow systems to classify, cluster and adapt to new threats dynamically[12]. Deep learning models also enhance the performance of detectors by helping in capturing more nuanced patterns in massive datasets. In this respect, this research paper suggests an intelligent machine learning structure to detect and monitor cybersecurity threats, and it can effectively identify malicious activities whilst being able to adapt to changing cyber-attack strategies.

1.1 Motivation and Contribution

The inability of conventional security measures to identify sophisticated threats is highlighted by the growing frequency and complexity of cyberattacks. This encourages the implementation of intelligent ML methodologies that can efficiently identify concealed patterns and analyze vast amounts of data. Thus, an adaptable and automated system of cybersecurity should be developed to detect the threat accurately and in real-time. This study has a number of important contributions as follows:

- Conducted an in-depth exploratory analysis of the ToN-IoT dataset, highlighting critical challenges such as class imbalance and skewed packet size distribution using visualization techniques.
- Improves minority class detection using SMOTE, increasing the reliability of the model in imbalanced classification scenarios.
- Proposed an integrated framework combining RNN and XGBoost for ensemble-based prediction, leveraging strengths of both ML and DL.
- Achieves superior performance across multiple evaluation metrics and outperforms traditional and deep learning models in cybersecurity threat detection.
- Demonstrates strong model generalization and stability, making the proposed approach suitable for real-world intrusion detection applications.

The growth of complexity and volume of the traffic of the IoT network requires sophisticated intrusion detecting mechanisms that can manage sequential patterns and unbalanced data distributions. Current methods tend to employ solitary machine learning or deep learning models, which restrict their capability to attain high accuracy, as well as, strong generalization. To overcome these shortcomings, this paper combines a RNN and XGBoost, which is a combination of time learning and ensemble forecasting. The originality of this piece is that it uses hybrid RNN-XGBoost architecture with effective preprocessing, SMOTE-based balancing, and feature selection that enhances its accuracy, balanced performance measures, and good generalization to the IoT cybersecurity application.

1.2 Organization of the Paper

This is how the paper is structured: The literature on cybersecurity threat detection and monitoring is reviewed in Section II; the dataset, pre-processing techniques, and model development are described in Section III; Section IV discusses the experimental outcomes and comparison, while the most important results and potential outcomes of the study are detailed in Section V.

2. Literature review

An overview and critical review of major research studies on cybersecurity threat detection and monitoring were conducted to inform and advance this research.

Sivasundaram et al. (2025) developed a novel framework for the early detection of unauthorized access by leveraging multimodal data fusion techniques using CNNs and LSTM networks. The proposed framework has a high level of operational accuracy of 93, good level of sensitivity and specificity in the detection of unusual access behavior and statistically significant results ($p<0.05$). The hybrid architecture helps to enhance the real-time responsiveness of the system and also makes a significant contribution to the development of intelligent cybersecurity systems that can be used to detect and notify of threats in real time[13].

Rongali (2025) proposed framework integrates machine learning-based anomaly detection techniques, supervised classification models, and behavioral analytics to enhance cybersecurity monitoring. Experimental outcomes demonstrate that the Neural Network model achieved the highest accuracy (96.3%) with the lowest false positive rate (2.9%), followed by Random Forest (94.5% accuracy, 3.8% false positives) and Isolation Forest (89.1% accuracy, 7.2% false positives)[14]. Lu, Wu and Chen (2024) propose a framework that integrates ML and DL to improve their liability and network security protection capabilities of industrial control systems. After the experiment, the accuracy of the improved algorithm is more than 85%, the prediction is more than 75, the regression line is more than 55%, and the modified algorithm is better than XGBost and other algorithms. The concentration of the improved CNN in identifying attack information can reach 95% [15].

Eswari and Lakshmi (2024) focus on efficiently monitoring and analyzing network traffic. The proposed system achieves notable advancements in detection accuracy, with results showing 98.7% accuracy on the NSL-KDD dataset and 97.5% on the CICIDS2017dataset. Moreover, the framework is set to be flexible, as well as responsive, being able to detect threats of DDoS much faster than the traditional methods, thus providing a powerful tool to network administrators to proactively handle DDoS threats[16]. Lin et al., (2022) suggested method may significantly improve the labeling attack strategy's effectiveness. The trial

result demonstrates that the approach's F1 score is more than 90% and up to around 96%, which may help cybersecurity specialists identify tactics and give a strong foundation for further alert correlation[17]. Dutta and Kant (2021) direct attention to the many security challenges faced by IoT devices and the required layer-by-layer security processes. In addition, the author integrated the TensorFlow module with the CTI platform to create a TinyML framework that can accurately predict threats to Smart Devices using a Supervised MLclassifier. The training dataset had a success rate of 96.8% and the test dataset had a success rate of 96.3% after applying the final solution[18].

Table I summarizes current studies on cybersecurity threat detection and monitoring, including the proposed models, datasets used, significant findings, and challenges faced

Table 1: Literature study on Cybersecurity Threat Detection using Machine learning approaches

Author	Data	Techniques	Results	Limitations & Future Work
Sivasundaram et al. (2025)	Multimodal cybersecurity data	Hybrid CNN-based framework for detecting unusual access behavior	93% accuracy, strong sensitivity & specificity ($p<0.05$)	Needs evaluation on large-scale real-time environments
Rongali (2025)	Network traffic/anomaly data	ML-based anomaly detection with supervised models and behavioral analytics	NN: 96.3% accuracy, 2.9% FPR; RF: 94.5%; Isolation Forest: 89.1%	Requires optimization for scalability and real-time deployment
Lu, Wu and Chen (2024)	Industrial control system data	Improved CNN algorithm for attack detection	>85% accuracy, ~95% attack identification	Limited evaluation across diverse datasets
Eswari and Lakshmi (2024)	NSL-KDD, CICIDS2017 datasets	Adaptive framework for DDoS detection	98.7% (NSL-KDD), 97.5% (CICIDS2017)	Limited focus on other attack types beyond DDoS
Lin et al. (2022)	Cybersecurity attack labeling data	Attack tactic labeling mechanism	F1-score up to 96%	Needs integration with real-time IDS systems
Dutta and Kant (2021)	IoT threat data	TinyML-based framework with Naïve Bayes classifier	96.8% (train), 96.3% (test) accuracy	Limited scalability for large and complex datasets

Research gaps: Although cybersecurity threat detection has made great strides, the models currently in use have limitations in terms of scalability and their application in real-time in large and dynamic systems. Most methods are restricted to particular kinds of attacks or data, making them less generalizable to a wide range of threats that are constantly changing. Also, some of the methods can hardly find a balance between high accuracy and low FP. These limitations illustrate the necessity of a more robust solution that can be used to capture more complex patterns and improve the detection overall in the real world.

3. Research Methodology

The proposed methodology employs the ToN-IoT dataset to detect cybersecurity threats and includes data pre-processing, normalization, feature selection as well as SMOTE for class balancing. A data split of 80:20 is used, and a hybrid model of RNN and XGBoost is used to capture both sequential and complex feature patterns. The following metrics are used to evaluate the model's performance: accuracy, precision, recall, F1score, and ROCcurve. Fig. 1 is a suggested flowchart for cybersecurity threat identification and monitoring made possible by ML.

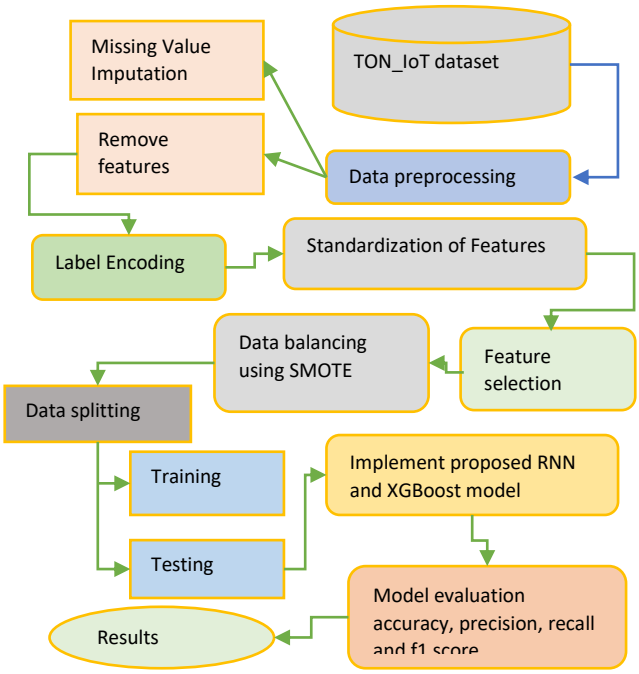

Fig. 1. Proposed Flowchart for Threat Detection And Monitoring

The suggested technique is described in depth in the following section:

3.1 Data Gathering and Analysis

In this paper, the ToN-IoT dataset, a large-scale intrusion detection dataset, has been analyzed. A sample of approximately 10,000 was used, but the entire dataset consists of millions of records from various sources. The analysis of the distribution of attacks and the correlation of features was performed with the help of data visualization tools like bar plots and heatmaps:

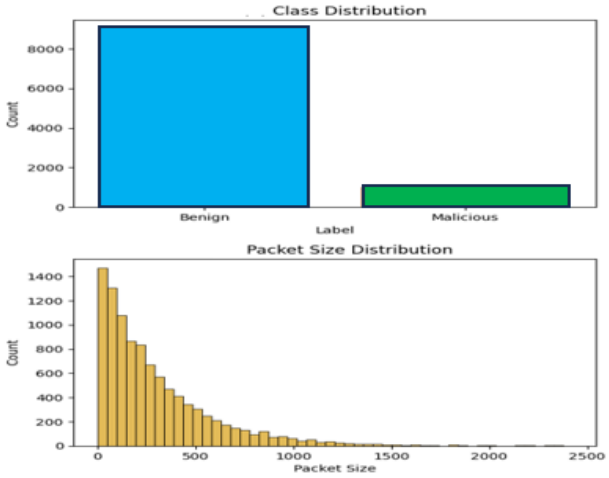

Fig. 2. Distribution of Class and Packet Size

Fig. 2 presents two key dataset characteristics. The first bar chart shows the class distribution, which shows that there are more benign samples (around 9,000) than harmful examples (about 1,000), indicating that the dataset is skewed. The second chart shows the distribution of the packet size, where smaller packets are more common, and the number of packets decreases with the size. Combining all these plots informs about traffic structure and highlights the issues of the class imbalance and distorted packet size distributions affecting model training and testing.

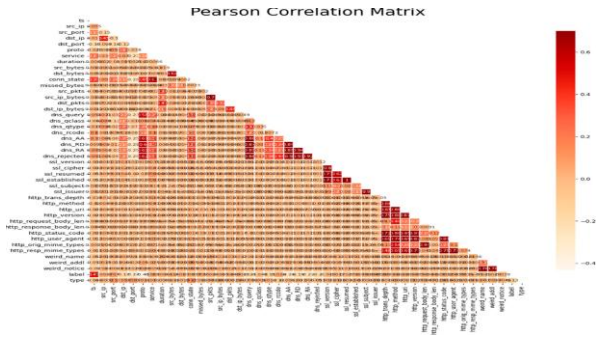

Fig. 3. Correlation Heatmap of Dataset Features

Fig. 3 displays the heatmap of the TON_IoT dataset, which depicts the correlation between features, with strong positive and negative correlations that are pertinent to the detection of cybersecurity threats. Redundancy, as shown by clusters of strongly correlated characteristics, is crucial when choosing features to improve the efficiency and performance of the model.

3.2 Data Pre-processing

Data preparation using the ToN-IoT dataset involved data concatenation, cleaning, and feature engineering. The pre-processing activities included handling missing data through imputation, ensuring data consistency and proper formatting, and labeling and normalizing the data. The most important pre-processing processes are outlined as follows:

- **Missing Value Imputation:** Use statistical methods to fill in data gaps and prevent misclassification.
- **Remove features**: Feature "src_ip" and feature "dest_ip" were omitted from the dataset since they do not affect the target variables.
- **Label Encoding:** The Label Encoding (LE) method was employed to convert categorical attributes into numerical values, enabling their use in ML models. Here, the distinct types of data in a feature received distinct integer values, so that all categorical data would be well represented and read by the model.

3.3 Standardization of Features (StandardScaler)

The StandardAero() function was used to normalize the dataset due to the various scales of the descriptors, resulting in a distribution with a mean of 0 and a Standard Deviation of 1. Equation (1) shows the process for achieving this transformation by dividing the total by the standard deviation and then removing the mean value of each observation:

$$z = \frac{x-\mu}{\sigma} \qquad (1)$$

the feature's Converted Value (z), the Original Values (x) of all the descriptors in the dataset, the mean (μ), and the StandardDeviation (σ).

3.4 Feature selection

The criteria for selecting the top 15 features with the K-Best algorithm are grounded in their profound impact on IoT security. The features are deemed more important, as they have an exceptionally high level of effective separation between normal and malicious actions within the IoT networks. The K-Best algorithm is an accurate means of finding these important features, which can help to build a more robust defense against emerging threats. Fig. 4 shows the scores of the feature importance obtained through the Select Best approach, which shows the variables that have the strongest impact on the predictive performance of the model. The plot indicates that timestamp (ts) has the highest score followed by protocol (proto) and destination port (dst_port) meaning that they are important aspects of classification. Additional characteristics like DNS-related attributes and source/destination identifiers are moderately contributing, and the impact of SSL and the weird notices is comparatively low. The ranking offers a clear understanding of the discriminative strength of every feature and informs the interpretability and possible feature selection to optimize the selection.

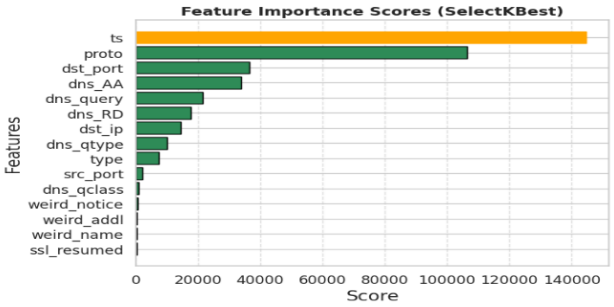

Fig. 4. Plot for Feature Importance Score with SelectkBest

3.5 Data balancing using SMOTE

For balancing the dataset, the SMOTE was utilized. This is applied to attack classes. The SMOTE technique mitigates overfitting by generating synthetic minority-class samples. The SMOTE pre-processing method was an early trailblazer for practitioners of imbalanced categorization within the scientific community. The SMOTE approach is highly esteemed in the fields of ML and data mining for its data pre-processing and sampling capabilities.

3.6 Data Splitting

An 80:20 split is used to separate the dataset into training and testing subsets, with 80% going toward model training and the remaining 20% set aside for performance assessment.

3.7 Implement of Models

In this work, a deep ML model is proposed for cybersecurity threat detection combining an RNN and XGBoost.

1) Proposed Recurrent Neural Network (RNN) Model

An RNN is a DL model specifically designed to process sequential and time-dependent data, such as network traffic. It is widely used in applications like natural language processing, intrusion detection systems due to its ability to learn patterns over sequences. The RNN takes into account both the current input and the previously calculated hidden state at each stage of its sequential processing of incoming data. As a result, the network is able to store data from previous time steps and develop contextual awareness throughout the sequence. The hidden state of the RNN is computed as Equation. (2)

$$h_t = tanh\,(W_x x_t + W_h h_{t-1} + b_{h)} \qquad (2)$$

The hidden state $\mathrm{h_t}$ is the internal memory of the network at time step t. It uses the recent input $\mathrm{x_t}$ an and the past hidden state $\mathrm{h_{t-1}}$ to enable the model to learn the temporal relationships in sequential data. The output of the RNN is given by Equation. (3):

$$y_t = W_y h_t + b_y \qquad (3)$$

The output y_t is generated from the hidden state and represents the final prediction or classification result. This output may be utilized in sequence prediction, classification or forecasting tasks depending on the application. The RNN is capable of retaining past information due to its recurring structure, but it can be problematic, e.g., the vanishing gradient problem can arise when working with long sequences. This limitation notwithstanding, it is a paradigm model in sequence modeling tasks. The RNN model uses 2 hidden layers with 128 units each, the tanh activation function and the Adam optimizer with a Learning Rate of 0.001. The Dropout Rate of 0.2 and the binary cross-entropy loss function are used to avoid overfitting and enhance classification results.

2) Extreme Gradient Boosting (XGBoost) Model

The XGBoost method is a kind of ensemble learning that relies on DT for prediction purposes. It may be employed in regression to minimize a loss function that quantifies the discrepancy between the actual and anticipated objective values.The mathematical framework for XGBoost regression is shown in Equation (4):

$$y = f(x) \qquad (4)$$

In this context, x represents the input characteristics vector (e.g., square footage, number of bedrooms), y represents the projected property price, and f(x) is the XGBoost model that uses x to predict y. When XGBoost needs to determine f(x), it builds a network of decision trees and trains them to reduce the mean squared error (MSE) lossfunction. The model generates a final forecast by combining predictions from many decision trees. The XGBoost regression model may be represented in its generic form as Equation (5):

$$y = \sum(k = 1\ to\ K)\ fk(x) \qquad (5)$$

where K is the sum of all the ensemble's decision trees and fk(x) represents their forecast. Weighted sums of the tree's leaf values learned during training make up each tree's forecast. The XGBoost model's prediction for a given input x is determined by summing the predictions of all the ensemble DTs. For the XGBoost model, 200 estimators are used with a learning rate of 0.1 and a maximum depth of 6 to optimize prediction accuracy. The subsample and colsample_bytree values are set to 0.8 to enhance generalization and reduce overfitting. Additionally, gamma is set to 0.1, min_child_weight to 1, with binary:logistic objective and logloss as the evaluation metric.

3.8 Evaluation metrics

There are several performance metrics that are used to test the effectiveness of a proposed design. To initially summarize the classification results, a ConfusionMatrix was created, which presented the number of correct and incorrect predictions per class. A model's performance is assessed using TP, TN, FP, and FN values. Besides these values, the performance of the model has been measured by using precision, recall, F1-score, and accuracy as well. Thus, have applied all of these measures to test our proposed model and compare our findings:

Accuracy: It is the percentage of instances in the dataset (input samples) for which the trained model produced an accurate prediction divided by the total number of instances in the dataset, as expressed in Equation (6)-

$$Accuracy = \frac{\mathrm{TP}+TN}{\mathrm{TP+FP+TN+FN}} \quad (6)$$

Precision: Precision is defined as the proportion of correctly predicted positive cases relative to all occurrences that the model projects as positive. It expressed in Equation (7)

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \quad (7)$$

Recall: The proportion of correctly anticipated positive cases compared to all positive occurrences is shown by this indicator. It is mathematically expressed in Equation (8)

$$Recall = \frac{\mathrm{TP}}{TP+FN} \quad (8)$$

F1 score: It is defined as the harmonic mean of prec and rec, which helps to balance both measures effectively. Its value ranges from 0 to 1, and it is mathematically expressed in Equation (9)

$$F1-score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (9)$$

The evaluation metrics are selected for their effectiveness in assessing model performance.

4. Results and Discussion

This section describes the experimental design and highlights the performance of the proposed model at various stages with particular focus on the results of the evaluation and the efficiency of the computation.

4.1 Experimental Setup

A high-performance system with a 2 TB NVMe SSD for fast data access, 64 GB of DDR4 RAM, an Intel Core i9-12900K 16-core CPU running at 3.2 GHz, and an NVIDIA GeForce RTX 3090 with 24 GB of GDDR6X memory is used for the research. The models were created and implemented using Python 3.9.7 in a software environment based on Ubuntu 20.04 LTS.

4.2 Evaluation Results

Table II shows the outcomes of the classification of the proposed models to detect and monitor cybersecurity threats on the ToN_IoT data set. The RNN model had an accuracy of 98.5, slightly less than that of XGBoost at 98.7 and also showed a higher precision and recall values. Compared to F1-score and AUC, RNN performs slightly better, indicating a balanced and robust classification ability. Overall, both models exhibit high effectiveness, with RNN providing slightly more consistent results across evaluation metrics.

Table 2: Results of proposed models for threat detection

Matrix	RNN	XGBoost
Accuracy	98.5	98.7
Precision	98.3	98.1
Recall	98.7	98.4
F1-score	98.5	98.2
AUC	98.8	98.7

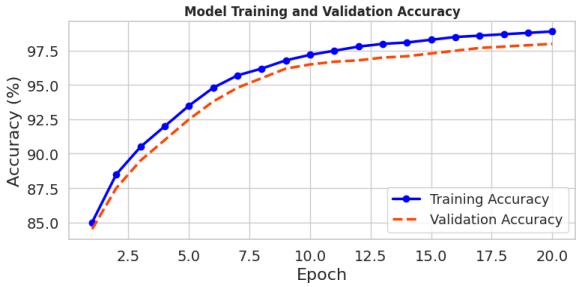

Fig 5: Accuracy Curve for the Proposed RNN Model

Fig. 5 demonstrates fast growth in the initial epochs, and then a slow plateau, meaning the successful learning and convergence of the model. The training curve is always slightly higher than the validation curve during the process. Their small and constant distance indicates that good generalization is possible with minimal overfitting.

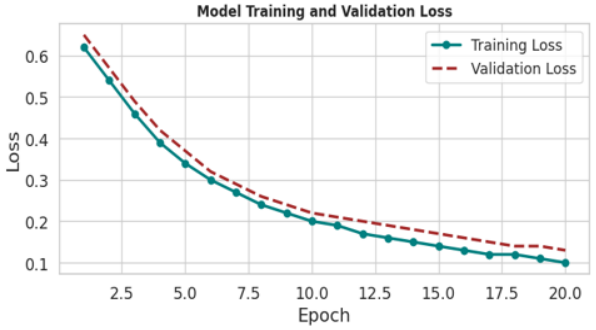

Fig. 5. Loss curve for the proposed RNN Model

Fig. 6 shows a sharp decrease in the initial epochs followed by a smooth, gradual decline, indicating effective learning and convergence of the RNN model. The validation loss is close to the training loss with a small gap, indicating a good overall generalization and no substantial overfitting.

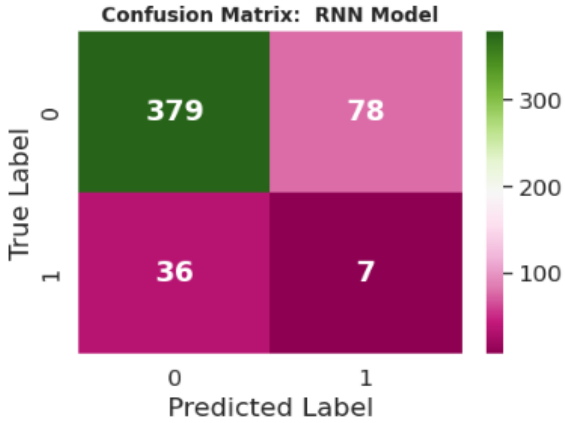

Fig. 6. Confusion Matrix for the Proposed RNN Model

Fig. 7 demonstrates that out of 379 True Negatives and 7 True Positives, the suggested RNN model performed well on the majority class but had trouble detecting the minority class. A lack of class balance, as seen by the 78 FP and 36 FN, affects the model's recall and precision in Class.

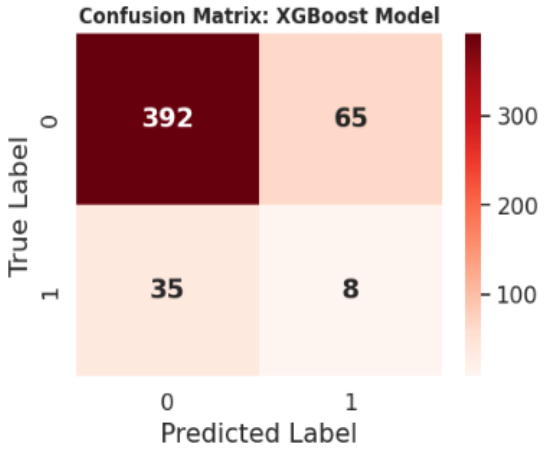

Fig. 7. Confusion Matrix for the Proposed XGBoost Model

Fig. 8 indicates that XGBoost model was able to identify 392 True Negatives and 8 True Positives, which indicates good performance in the MajorityClass with poor performance in the MinorityClass. The class imbalance, with 65 FP and 35 FN, results in poorer precision and recall on Class 1 in spite of the high overall accuracy.

4.3 Comparative Analysis

The comparative accuracy analysis of the models with current models is provided in Table III. The table emphasizes the effectiveness of different ML and DL methods to detect and monitor cybersecurity threats. Traditional models like Naïve Bayes (NB) have lower accuracy (80.4%), but DL models like LSTM and CNN can significantly improve on the result. RF has a rather high level of accuracy, but lower precision and recall values. In contrast, the proposed models outperform all others, indicating superior and well-balanced performance across all evaluation metrics.

Table 3 : Comparison of Different ml and dl Models for threat detection and monitoring

Model	Accuracy	Precision	Recall	F1-score
NB[19]	80.4	87.4	80.4	81.4
LSTM[20]	87	89	88	88
CNN[21]	97.4	95	96	95
RF[22]	97.8	87.5	85.4	86.4
RNN	98.5	98.3	98.7	98.5
XGBoost	98.7	98.1	98.4	98.2

The suggested method provides great detection accuracy and balanced performance on various evaluation metrics and is effective in combining RNN and XGBoost. It improves the reliability of the model by employing solid pre-processing, feature selection, and balancing the data using SMOTE. Temporal patterns and intricate interactions between features are captured by the framework and enhanced threat detection capability is achieved. Also, it shows good generalization and with little overfitting thus it can be applied in practice in IoT cybersecurity.

5. Conclusion and future study

Cybersecurity threats are dynamic, and conventional rule-based defenses are insufficient to safeguard sensitive information and cyber infrastructure because they cannot identify new or disguised attacks. To address this threat, Cyber Shield proposes an AI-based platform that uses DL and ML algorithms to identify and classify threats precisely. The proposed method combines RNN with XGBoost, which is effective in capturing both sequential and non-sequential patterns of feature relationships in network traffic. Experimental evidence shows that the proposed models are highly accurate and balanced in their evaluation metrics and outperform a number of existing ML and DL models. Additionally, the models have good generalization and less overfitting, which implies that they are reliable in real-world applications. Altogether, this paper demonstrates the efficiency of hybrid learning models to combat current cybersecurity issues and offers a bright perspective on the development of the future Internet of Things IDS.

5.1 Limitations and Future Work

Although the suggested model achieves a high degree of accuracy, it may only be applicable to the circumstances of a large-scale, real-time, IoT environment as it is evaluated on a sampled fraction of the dataset. This is despite the fact that class imbalance might still be observed after SMOTE, which still impacts minority class detection. Moreover, the model does not consider the computational complexity and real-time deployment limitations entirely. To ensure the approach is tested on the full dataset in the future, it is possible to consider supporting the approach with more sophisticated architectures (e.g., LSTM or attention-based models) and streamlining the framework to support real-time intrusion detection at a lower computational cost.

Funding: This research received no external funding

Conflicts of Interest: The authors declare no conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers.

References

[1] Aiyanyo, I.D., Samuel H and Lim, H. (2020). A Systematic Review of Defensive and Offensive Cybersecurity with Machine Learning," *Appl. Sci.*, vol. 10, no. 17, p. 5811,, doi: 10.3390/app10175811.

[2] Almotairi, A and Khafajah, N.M. (2024). "Enhancing intrusion detection in IoT networks using machine learning-based feature selection and ensemble models," *Syst. Sci. Control Eng.*, vol. 12, no. 1, 2024, doi: 10.1080/21642583.2024.2321381.

[3] Al-Taleb N and Saqib, N. (2022). "Towards a Hybrid Machine Learning Model for Intelligent Cyber Threat Identification in Smart City Environments," *Appl. Sci.*, vol. 12, no. 4, p. 1863, Feb. 2022, doi: 10.3390/app12041863.

[4] Bhattacharjee D. (2025) "Design and Evaluation of Deep Generative AI Model for Intrusion Detection in Cyber Threat Monitoring," in *2025 7th International Symposium on Advanced Electrical and Communication Technologies (ISAECT)*, 2025, pp. 1–6. doi: 10.1109/ISAECT68904.2025.11318752.

[5] Dutta A and Kant, S. (2021). Implementation of Cyber Threat Intelligence Platform on Internet of Things (IoT) Using TinyML Approach for Deceiving Cyber Invasion," in *International Conference on Electrical, Computer, Communications and Mechatronics Engineering, ICECCME 2021*, 2021. doi: 10.1109/ICECCME52200.2021.9590959.

[6] Eswari D.S and Lakshmi, P.V. (2024). Advanced Detection of DDoS Attacks Using Hybrid Deep Learning Models and Federated Learning," in *2nd International Conference on Cyber Physical Systems, Power Electronics and Electric Vehicles (ICPEEV)*, 2024, pp. 1–6. doi: 10.1109/ICPEEV63032.2024.10932071.

[7] Gupta, S and Passerone R. (2023) "An Investigation of Cyber-Attacks and Security Mechanisms for Connected and Autonomous Vehicles," *IEEE Access*, vol. 11, pp. 90641–90669, 2023, doi: 10.1109/ACCESS.2023.3307473.

[8] Hamid I and Rahman M. M. H. (2025). AI, machine learning and deep learning in cyber risk management, *Discov. Sustain.*, vol. 6, doi: 10.1007/s43621-025-01012-3.

[9] Jain D and Jain, S. (2026). Artificial Intelligence (AI)-Driven Network Traffic Anomaly Detection for IT Infrastructure Security, IEEE, 2026, pp. 1–6. doi: 10.1109/ICAIC67076.2026.11395685.

[10] Khan N.W *et al. (2023).* "A hybrid deep learning-based intrusion detection system for IoT networks," *Math. Biosci. Eng.*, vol. 20, no. 8, pp. 13491–13520, 2023, doi: 10.3934/mbe. 2023602.

[11] Lin S.X., Chen T.Y and Wu D.J. (2022). Attack Tactic Labeling for Cyber Threat Hunting," in *2022 24th International Conference on Advanced Communication Technology (ICACT)*, IEEE, Feb. 2022, pp. 34–39. doi: 10.23919/ICACT53585.2022.9728949.

[12] Lu, X., Wu K and Chen, Y. (2024) Enhancing Situational Awareness in Industrial Automation and Cybersecurity Through an Integrated Framework That Leverages Both Machine Learning and Deep Learning Technologies," in *2024 IEEE 5th International Conference on Pattern Recognition and Machine Learning (PRML)*, IEEE, Jul. 2024, pp. 41–46. doi: 10.1109/PRML62565.2024.10779917.

[13] Manda, J.K. (2024). AI-powered Threat Intelligence Platforms in Telecom: Leveraging AI for Real-time Threat Detection and Intelligence Gathering in Telecom Network Security Operations, *SSRN Electron. J.*, 2024, doi: 10.2139/ssrn.5003638

[14] Nutalapati, P., Vummadi, J.R Dodda S and Kamuni, N. (2025). Advancing Network Intrusion Detection: A Comparative Study of Clustering and Classification on NSL-KDD Data, in *2025 International Conference on Data Science and Its Applications (ICoDSA)*, IEEE, Jul. 2025, pp. 880–885. doi: 10.1109/ICoDSA67155.2025.11157595.

[15] Okafor, M. O. (2024). Deep learning in cybersecurity: Enhancing threat detection and response, *World J. Adv. Res. Rev.*, vol, pp. 1116–1132, Dec. 2024, doi: 10.30574/wjarr. 2024.24.3.3819

[16] Rongali, S.K (2025). AI-Powered Threat Detection in Healthcare Data," in *2025 International Conference on Artificial Intelligence and Machine Vision (AIMV)*, IEEE, Aug. 2025, pp. 1–7. doi: 10.1109/AIMV66517.2025.11203733.

[17] Siale, A.Y.D., Hassan, Q.M.Z and Veena, B.S. (2025). Enhancing Large-Scale Network Security with a VGG-Net-Based DCNN: A Deep Learning Approach to Anomaly Detection," *J. Robot. Control*, vol. 6, no. 3, pp. 1316–1331, 2025, doi: 10.18196/jrc.v6i3.25169.

[18] Singh S. (2025). Advancing Network Security in 5G: Leveraging the 5G-NIDD Dataset for Intrusion Detection and Mitigation," in *2025 IEEE 12th International Conference on Cyber Security and Cloud Computing (CSCloud)*, IEEE, Nov. 2025, pp. 1–6. doi: 10.1109/CSCloud66326.2025.00055.

[19] Sivasundaram, M., Hemalatha, S (2025). "Real-Time Threat Detection Using a Hybrid Machine Learning Algorithm for Enhanced Security," in *2025 4th International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, IEEE, Dec. 2025, pp. 348–352. doi: 10.1109/ICAAIC64647.2025.11331203

[20] Suparman, A., Akhmad EPA and Dinata B.M (2024). Leveraging Artificial Intelligence for Enhancing Cybersecurity: A Deep Learning

Approach to Real-Time Threat Detection, *J. Acad. Sci.*, vol.. 835–842, Nov. 2024, doi: 10.59613/0yv79c49.

[21] Tulsyan, R., Shukla, P T and Bhardwaj, A. (2024). Cyber Security Threat Detection Using Machine Learning," *INTERNATIONAL J. Sci. Res. Eng. Manag.*, vol, pp. 1–6, oct. 2024, doi: 10.55041/IJSREM37949.

[22] Yang L and Shami, A. (2023). Towards Autonomous Cybersecurity: An Intelligent AutoML Framework for Autonomous Intrusion Detection," in *Proceedings of the Workshop on Autonomous Cybersecurity*, New York, NY, USA: ACM, Nov. 2023, pp. 68–78. doi: 10.1145/3689933.3690833