Journal of Global Research in Multidisciplinary Studies (JGRMS)
Volume 2 | Issue 11 | November 2026 | Pages 15–22
DOI: https://doi.org/xxxxxxxxxxxxx
e-ISSN: 3049-2963
RESEARCH ARTICLE OPEN ACCESS
Available online: https://saanvipublications.com/journals/index.php/jgrms

An Autoencoder-Based Network Anomaly Detection Framework for Unknown Cyberattacks

Dr. Bal Krishna Sharma
Professor
Department of Computer Science and Application
Mandsaur University, Mandsaur
drbks7426@gmail.com

Corresponding Author: * Bal Krishna Sharma (drbks7426@gmail.com)

Received: 08 November 2026 | **Revised:** 12 November 2026 | **Accepted:** 22 November 2026

***Abstract*—As network attacks evolve at a rapid rate, advanced Network Intrusion Detection Systems (NIDS) that use machine learning and deep learning methods have become critical for detecting new and previously unknown cyber threats. Traditional detection strategies often fail to identify zero-day attacks and complicated malicious network actions owing to their dependency on predetermined signatures and acknowledged attack patterns. This research suggests a framework for detecting network anomalies in order to prevent unknown cyberattacks utilizing the TON_IoT dataset, which has around 46,000 records and 45 characteristics, in order to overcome these obstacles. Data cleansing, label encoding, Min-Max normalization, SMOTE-ENN class balance, and Select Best feature selection are all parts of the suggested framework. After that, CNN and RF models are implemented. A 99.97% F1-score, 99.98% recall, and 99.97% precision were achieved by the RF model, whereas a 99.99% F1-score, 99.98% recall, and 99.97% accuracy were achieved by the CNN model, according to the experimental data. These results show that ML/DL techniques may work together to improve network anomaly detection and help identify harmful network behaviors.**

***Keywords*—Deep Autoencoder, Zero-Day Attack, Unsupervised Learning, Intrusion Detection, Network Security, Anomaly Detection.**

I. INTRODUCTION

Cyberattacks against digital infrastructures have become more common due to the exponential expansion of computer networks, cloud computing, and the IoT [1]. Data security and network operations may be affected by regular threats such as brute-force assaults, denial-of-service (DoS) attacks, botnet activities, port scanning, web attacks, and penetration efforts. As a result, safeguarding digital resources and spotting fraudulent activity requires reliable network anomaly detection [2][3][4].

Conventional security measures, such as firewalls and NIDSs that use signatures to detect intrusions, mostly depend on already established rules and patterns of attacks [5]. These approaches often struggle to identify zero-day exploits and previously unseen cyberattacks because unfamiliar threats may not match existing signatures [6][7][8][9]. Consequently, advanced detection techniques are needed to recognize abnormal network behavior and improve the identification of emerging security threats [10].

Anomaly detection identifies deviations from expected patterns in network traffic and system activities [11]. Deep learning models called autoencoders (AEs) train to rebuild input features from compact data representations using an encoder and a decoder. Autoencoders often have smaller reconstruction errors for known patterns when trained on typical network data; nevertheless, they may create larger errors for suspicious behaviors, which can be utilized to detect possible cyberattacks [12][13][14].

In order to analyses traffic patterns, detect suspicious activity, and prevent cyberattacks, network security professionals often use Deep Learning (DL) and Machine Learning (ML) methods. In contrast to DL models, which learn intricate and nonlinear representations automatically from massive amounts of network data, ML algorithms may use extracted features to categories network behaviors [15][16]. The detection capabilities, support for identifying anomalous patterns, and analysis of high-dimensional data are all improved by these approaches. Anomaly detection, reliance on preset attack signatures, and the discovery of new and unknown cyber threats may all be enhanced by the integration of ML and DL techniques [17][18][19].

A. Motivation and Contribution

The growing sophistication of cyberattacks, coupled with the inadequacy of conventional intrusion detection systems to spot new types of threats, is what prompted this research. Current security measures aren't always up to the task of identifying new types of attacks or developing threats. To analyses network traffic and detect aberrant behaviors, effective ML and DL approaches are necessary. An improved cybersecurity and threat detection framework using convolutional neural network (CNN) and RF models is the goal of this research. Here are a few important points that this study brings up:

- Utilizes the TON_IoT dataset for network anomaly detection and cybersecurity analysis.

Copyright © 2026 The Author(s).
This is an open access article distributed under the terms of the Creative Commons Attribution 4.0 International (CC BY 4.0) License (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, adaptation, and reproduction in any medium, provided the original author(s) and the source are properly credited.

DOI: https://doi.org/xxxxxxxxxxxx

- Applies data preprocessing techniques to improve data quality and consistency.
- Uses SMOTE-ENN to address class imbalance and improve minority-class detection.
- Implements Select Best to identify relevant network traffic features.
- Develops CNN and RF models to classify normal and anomalous network activities.
- Evaluates model performance using accuracy, precision, recall, and F1-score.

B. Justification and Novelty

Traditional network intrusion detection approaches fail to detect new and emerging assaults; this is something that the suggested architecture aims to rectify. Data quality, class balance, and feature relevance are all enhanced by integrating data preprocessing, SMOTE-ENN, and Select Best. Combining CNN with RF allows for the investigation of intricate network traffic patterns via the use of DL and ML techniques. The framework offers a methodical way to differentiate between typical and unusual network behaviors. What makes it unique is how it brings together various methods to provide a cohesive framework for TON_IoT network anomaly detection.

C. Paper Structure

The following is the outline of the paper: Cyberattack detection and network anomaly detection have been the subject of previous research, which is reviewed in Section II. Section III outlines the dataset, the approaches used for data preparation, and the creation of the suggested model. The results of the experiments and comparisons of the performances are presented in Section IV. As a conclusion, Section V provides a concise overview of the key findings and suggests avenues for further study.

II. LITERATURE REVIEW

The suggested approach was built upon a detailed review of previous research on unknown cyberattack detection. Table I provides a concise overview of current studies on autoencoder-based anomaly detection, showcasing the methodologies, datasets, important results, and constraints of earlier techniques.

In their study, Bhushan et al. (2026) provide an autoencoder model–based neural network technique for real-time network anomaly detection driven by artificial intelligence. The technology observes standard traffic behavior and identifies irregularities that signify potential cyber-attacks. The suggested model is a good basis for adaptive intrusion detection systems as, according to the experimental findings, it accurately identifies anomalous patterns of network traffic. Experiments validate the efficacy of the suggested autoencoder-based model for real-time anomaly detection, showing a detection accuracy of 95.2% and a false positive rate of 4.3% [20].

Bachar, Khiat, and Guemmat (2026) provide a multi-tiered hybrid architecture for anomaly detection tailored for real-time IoT networks. They provide a method that integrates with several frameworks, beginning with data pretreatment, feature reduction, and lightweight anomaly detection. In order to implement a dual-stage detection technique, a hybrid CNN-LSTM model is constructed. In order to merge the results of the two detection phases, a decision fusion process is implemented. That is shown by the outcomes. On the CICHoT 2025 dataset, they reach a 98.6% accuracy rate with a detection delay of 29 ms [21].

In their study, Rani et al. (2025) used sequential models for predictive power and autoencoders for novelty detection to obtain very high detection accuracy with almost no false positives. The system functions in real-time due to the integration of edge computing, enabling traffic data preprocessing at designated points, swift model inference, and reduced latency. The benchmark datasets and associated real network traffic simulators were tested on, including NSL-KDD, CICIDS2017, and UNSW-NB15. Experiments have shown that although the average false positive rate is less than 2%, the detection accuracy of known assaults may reach over 98% and new threats can reach over 95% [22].

Udayakumar, Ragothaman, and Khare (2025) provide an ensemble framework for anomaly detection that incorporates Quantum Clustering (QIC), Autoencoder, Graph Learning, Hidden Markov Model (HMM), Hyperdimensional Computing, and Recurrence Quantification. When compared to conventional individual ML algorithms, the detection accuracy is much improved using an ensemble-based technique that combines the aforementioned algorithms in a structured flow. With over 90% accuracy, 94% precision, 92% recall, and 93% F1-score, the resulting ensemble model outperforms traditional ML and DL techniques [23].

Udayakumar, Ragothaman, and Khare (2025) provide an ensemble framework for anomaly detection that incorporates Quantum Clustering (QIC), Autoencoder, Graph Learning, Hidden Markov Model (HMM), Hyperdimensional Computing, and Recurrence Quantification. When compared to conventional individual ML algorithms, the detection accuracy is much improved using an ensemble-based technique that combines the aforementioned algorithms in a structured flow. With over 90% accuracy, 94% precision, 92% recall, and 93% F1-score, the resulting ensemble model outperforms traditional ML and DL techniques [24].

Vemula et al. (2024) assert that uncovering abnormal network traffic is vital, with the NSL-KDD dataset being acknowledged for its considerable network data. It's an ideal spot to get started. An adversarial generative network (GAN) based anomaly detection system capable of detecting suspicious network traffic patterns indicative of possible infiltration is the target of this study. The GAN cybersecurity approach is data-driven and flexible, unlike traditional techniques. It can adapt to new threats dynamically with an astounding purity of 96%. The algorithms used in GAN include deep learning models (90−94%), unsupervised machine learning (92%), hybrid genetic methods (88%), and classical NSL-KDD-based methods (83%) [25].

Almeida et al. (2023) present an autoencoder-based solution trained in a semi-supervised manner to detect anomalies in network traffic. The autoencoder is trained on normal network traffic, allowing the model to learn a compact representation of the regular traffic data. The model uses the reconstruction error as a mechanism for identifying anomalies in the network traffic. The proposed solution is designed to be efficient in terms of computational resources. The experiments were performed using the NSL-KDD dataset, a widely used benchmark for intrusion detection systems, and achieved an accuracy of 90.49% and an Fl-score of 91.81%, outperforming state-of-the-art methods in the same dataset [26].

DOI: https://doi.org/xxxxxxxxxxxx

Research gaps: Existing network anomaly detection approaches demonstrate promising performance but still face challenges in identifying previously unseen cyberattacks and evolving threat patterns. There are high computational requirements in some methods, and there can be higher false positive rates or latency for detection in other methods. Additionally, the efficacy of available models can differ from one dataset or network to another. Additionally, the reliability of detection and classification might be impacted by class imbalance and irrelevant features. Therefore, an efficient framework integrating data preprocessing, feature selection, and ML and DL models is needed to improve network anomaly detection using the TON_IoT dataset.

TABLE I. RECENT STUDIES ON NETWORK INTRUSION DETECTION FRAMEWORK FOR UNKNOWN CYBERATTACKS

Author	Proposed Work	Dataset	Key Findings	Limitations & Future Work
Bhushan et al. (2026) (Source)	Autoencoder-based real-time network anomaly detection.	CICIDS2017 Dataset	95.2% detection accuracy and 4.3% false positive rate.	Dataset validation and detection of diverse unknown cyberattacks require further investigation.
Bachar, Khiat and Guemmat (2026) (Source)	Hybrid CNN–LSTM framework with multistage anomaly detection and decision fusion.	CICIoT 2025	98.6% accuracy with 29 ms detection latency.	Further evaluation on unseen attacks and diverse network environments is needed.
Rani et al. (2025) (Source)	Sequential deep learning and autoencoder-based novelty detection with edge computing.	NSL-KDD, CICIDS2017, UNSW-NB15, and simulated live traffic.	Over 98% accuracy for known attacks, over 95% for novel threats, and less than 2% average false positive rate.	Broader real-world deployment and scalability testing remain important.
Udayakumar, Ragothaman and Khare (2025) (Source)	Ensemble anomaly detection combining autoencoders, graph learning, quantum clustering, and other techniques.	UNSW-NB15	Approximately 90% accuracy, 94% precision, 92% recall, and 93% F1-score.	The ensemble's computational complexity and generalization to unknown attacks require further evaluation.
A and Ravindran (2025) (Source)	Cloud-hosted adaptive intrusion detection using ML and DL models.	Custom IIoT Network Traffic Dataset	Up to 97% detection accuracy using SVM.	Evaluation of diverse attack types and previously unseen cyberattacks is needed.
Vemula et al. (2024)	Generative Adversarial Network (GAN)-based network anomaly detection to identify unusual traffic patterns.	NSL-KDD	Reported 96% detection performance, as described in the supplied summary.	Further validation against previously unseen cyberattacks and evaluation of autoencoder-based methods are needed.
Almeida et al. (2023) (Source)	Semi-supervised autoencoder using reconstruction error to detect network traffic anomalies.	NSL-KDD	90.49% accuracy and 91.81% F1-score.	Further testing on modern network datasets and previously unseen attacks is needed.

III. RESEARCH METHODOLOGY

The proposed methodology uses the TON_IoT dataset for network anomaly detection, followed by data cleaning, label encoding, and Min–Max normalization. SMOTE-ENN addresses class imbalance, while Select Best identifies relevant features, and stratified sampling divides the dataset into 75% training and 25% testing subsets. Finally, Random Forest (RF) and Convolutional Neural Network (CNN) models classify normal and anomalous network activities, with performance evaluated using accuracy, precision, recall, F1-score, and ROC analysis. Figure 1 illustrates the proposed flowchart for Network Intrusion Detection Framework.

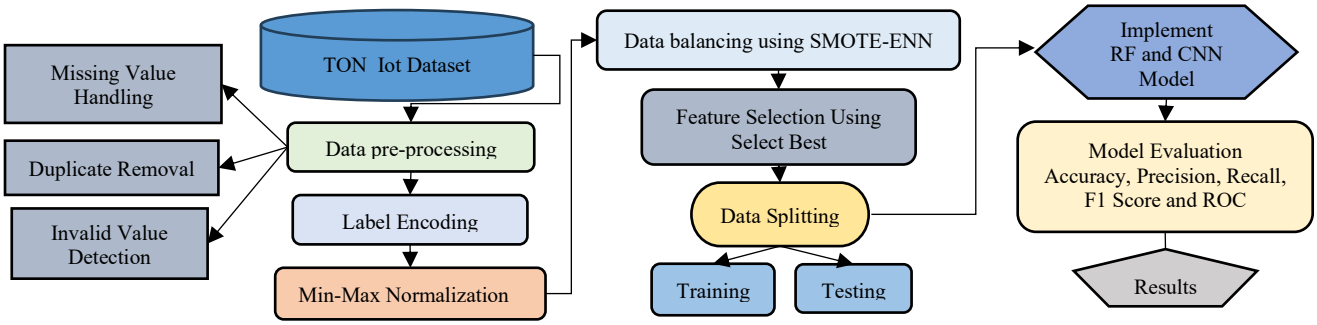

Fig. 1. Workflow of the Proposed Autoencoder-Based Cyberattack Detection Framework

The following section describes each stage of the proposed methodology in detail:

A. Dataset Collection and Exploratory Analysis

This study employs the TON_IoT dataset, containing approximately 46,000 rows and 45 features, to develop a framework for detecting previously unseen cyberattacks through network anomaly analysis. Bar plots and heatmaps are used to examine attack distributions, feature correlations, and relationships among network traffic attributes:

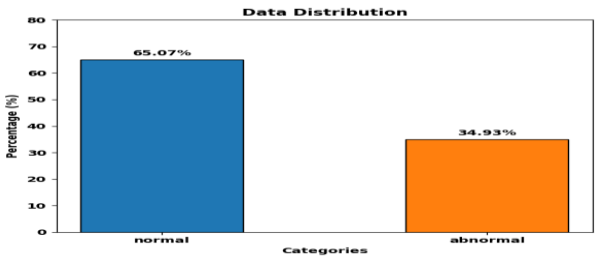

Fig. 2. Bar Graph of Class Distribution

J. Glob. Res. Multidiscip. Studies, 2 (11):15-22
DOI: https://doi.org/xxxxxxxxxxxx

Figure 2 illustrates the distribution of normal and abnormal data in the dataset using a pie chart. The normal category represents 65.07% of the data, while the abnormal category accounts for 34.93%. This distribution highlights the relative proportions of both classes and provides an overview of the dataset's class composition.

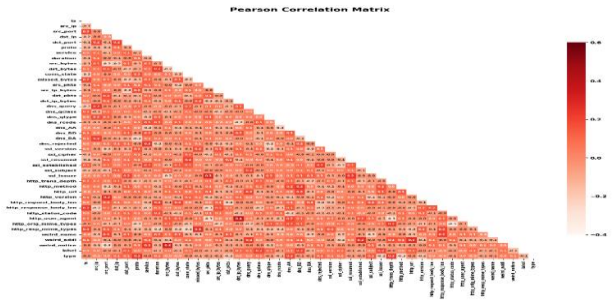

Fig. 3. Correlation Matrix Heatmap

Figure 3 illustrates the pairwise linear relationships among different network and protocol features using a lower-triangular layout with a red color gradient. The correlation coefficients range from negative to positive values, with some exceeding 0.6, indicating varying strengths of association between features. This visualization helps identify potential multicollinearity, redundant variables, and significant dependencies among network traffic attributes. Overall, the heatmap supports a better understanding of feature relationships and assists in selecting relevant attributes for network traffic classification.

B. *Data Pre-Processing*

The TON_IoT dataset is prepared through data integration, cleaning, and feature engineering. Preprocessing involved handling missing values, removing duplicate and redundant records, detecting invalid entries, applying label encoding, and normalizing numerical features. The main data preprocessing steps are summarized as follows:

C. *Data Cleaning*

Data cleaning is performed on the TON_IoT dataset using Pandas and NumPy to improve data quality and consistency. Missing values are replaced with feature-wise medians where necessary, while duplicate records are removed to eliminate redundancy. Infinite values, incomplete records, and invalid entries are also identified and handled to ensure reliable model training.

D. *Label Encoding*

Label encoding is a preprocessing technique used to convert categorical features into numerical values. It assigns a unique integer to each distinct category within a feature. This transformation enables machine learning models to process categorical data efficiently.

E. *Min-Max Normalization*

The normalization of records is carried out using the min–max technique to confine values within a range between 0 and 1. This is performed with the aim of optimizing the performance of the classifiers used and mitigating the effect of outliers. Normalization is conducted according to the following mathematical Equation (1):

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

where X represents the original value of the feature, X' is the normalized value, X_{min} is the minimum value of the feature, and X_{max} is the maximum value of the same.

F. *Data Balancing using SMOTE-ENN*

Data balancing is a preprocessing technique used to address class imbalance by adjusting the distribution of samples among different categories. It helps prevent predictive models from becoming biased toward the majority class and improves the identification of minority-class instances. SMOTE-ENN combines the Synthetic Minority Over-sampling Technique (SMOTE) with Edited Nearest Neighbors (ENN) to balance the dataset. SMOTE generates synthetic minority-class samples, while ENN removes noisy or incorrectly classified samples based on their nearest neighbors.

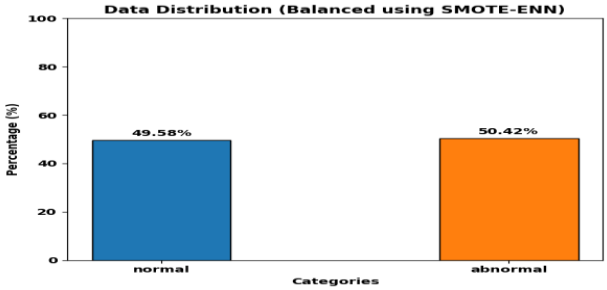

Fig. 4. Bar Graph of Class Distribution After SMOTE-ENN

Figure 4 presents the class distribution after applying the SMOTE-ENN resampling technique. The normal class is 49.58 per cent of the data, while the abnormal class is 50.42 per cent of the data. The proportions suggest that SMOTE-ENN successfully achieves class balance, which helps overcome class imbalance and facilitates the learning of more reliable models.

G. *Feature Selection using Selectbest*

Feature selection is a data preprocessing technique used to identify the most relevant features from a dataset for improving model performance. It reduces dimensionality by eliminating irrelevant or less informative attributes while retaining significant information. Select Best is a feature selection method that ranks features according to their statistical scores and selects the top-ranked features. It helps simplify the dataset, reduce computational complexity, and improve learning efficiency. In this study, Select Best was applied to identify the most significant network traffic features based on their calculated scores. As shown in Figure 5, the horizontal bar chart titled “Top Features Selected by Select Best” presents the scores of fifteen selected network traffic features, including ts, src_port, dst_ip, dst_port, and proto. Among these, ts, proto, and dst_port exhibit comparatively higher scores, indicating their greater statistical relevance, while the remaining features show considerably lower scores.

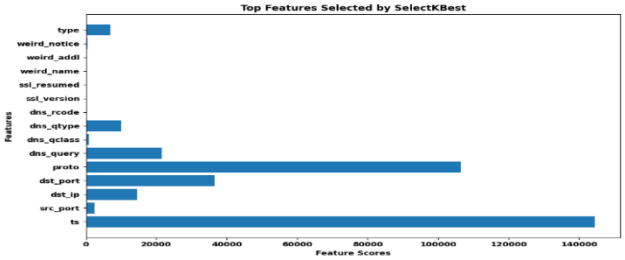

Fig. 5. Top Features Selected by Selectkbest

DOI: https://doi.org/xxxxxxxxxxxxx

H. Data Splitting

The dataset divided into training and testing subsets using stratified sampling with a split ratio of 75:25. This approach maintained the original class proportions in both subsets, ensuring representative data distribution and reliable model evaluation.

I. Implementation of the Proposed RF and CNN Models

In this work, two supervised learning models, Random Forest (RF) and Convolutional Neural Network (CNN), are employed to develop a Network Anomaly Detection Framework for Unknown Cyberattacks. RF identifies malicious network activities through ensemble-based decision-making, while CNN extracts relevant traffic patterns to distinguish normal behavior from anomalous activities.

1) Random Forest (RF) Model

Random Forest (RF) is an ensemble learning algorithm that combines multiple decision trees to improve classification performance and reduce overfitting. Each tree is trained using a randomly selected subset of training samples and features, and the final prediction is determined through majority voting. The classification output of the RF model is expressed in Equation (2).

$$\hat{y} = mode\{T_1(x), T_2(x), \dots\dots, T_N(x)\} \quad (2)$$

where $\hat{y}$ denotes the predicted class, x represents the input feature vector, $T_i(x)$ is the prediction of the i-th decision tree, and N represents the total number of trees. The majority-voting mechanism combines individual tree predictions to generate the final classification result. The importance of each feature can also be estimated using the mean decrease in impurity across the trees. Equation (3) defines the feature importance based on the weighted impurity reduction.

$$FI_j = \frac{1}{N}\sum_{i=1}^{N}\sum_{t\in T_i}\Delta I_{t,j} \quad (3)$$

where FI_j represents the importance of feature j, and $\Delta I_{t,j}$ denotes the weighted impurity reduction associated with that feature at node t. This process helps identify influential network traffic attributes and supports the interpretation of classification decisions. The RF model is configured with 100 decision trees, a maximum depth of 20, a minimum of 2 samples for splitting, and 1 sample per leaf. The model uses the Gini criterion, square-root feature selection, bootstrap sampling, balanced class weights, a random state of 42, and all available processor cores for parallel processing.

2) Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) is a deep learning model that extracts meaningful patterns from input data through convolutional filters. In network anomaly detection, a CNN can learn local relationships among traffic features to distinguish normal network activities from malicious or anomalous behavior. The convolution operation used to extract feature maps is represented in Equation (4).

$$Z_{i,j}^{(k)} = f\left(\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} W_{m,n}^{(k)} X_{i+m,j+n} + b^{(k)}\right) \quad (4)$$

where X represents the input feature matrix, $W^{(k)}$ denotes the convolutional kernel, $b^{(k)}$ is the bias term, f is the activation function, and $Z^{(k)}$ represents the resulting feature map. The convolution layer identifies relevant patterns that are passed to subsequent layers for further processing.

After feature extraction, the CNN uses a fully connected layer and a classification function to produce class probabilities. For binary classification, the output probability can be calculated using Equation (5).

$$P(y = 1|X) = \sigma(w^T h + b) = \frac{1}{1+e^{-(w^T h+b)}} \quad (5)$$

where h represents the extracted features, w denotes the output-layer weights, b is the bias, and σ is the sigmoid function. The resulting probability is compared with a predefined threshold to classify the input as normal or anomalous. The CNN model is configured with 50 epochs, a batch size of 32, a learning rate of 0.001, and the Adam optimizer. It consists of two convolutional layers with 32 and 64 filters, a kernel size of 3, ReLU activation, a sigmoid output activation, a dropout rate of 0.2, binary cross-entropy loss, and a validation split of 0.2.

J. Evaluation Metrics

A set of standard classification measures are used to assess the performance of the proposed model. To provide a representation of the true and false predictions for each class, a confusion matrix is generated, which includes True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). These values are then used to calculate accuracy, precision, recall, and F1-score to assess the model's classification performance:

$$Accuracy = \frac{\text{TP+TN}}{\text{TP+Fp+TN+FN}} \quad (6)$$

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

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

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

The performance of the proposed model is evaluated using several metrics. Accuracy measures the proportion of correctly classified instances among all samples, while Precision indicates the proportion of correctly predicted positive instances among all predicted positives. Recall measures the proportion of actual positive instances correctly identified by the model, and the F1-score represents the harmonic mean of precision and recall, balancing both metrics within a range of 0 to 1. The Receiver Operating Characteristic (ROC) curve illustrates the relationship between the True Positive Rate (TPR) and False Positive Rate (FPR) across different classification thresholds, where TPR represents sensitivity and FPR equals one minus specificity. The mathematical Equations are shown in (6) to (9).

IV. Results and Discussion

This section presents the experimental configuration and evaluates the proposed model's performance, highlighting its effectiveness and computational efficiency.

A. Experimental Setup

The proposed model is evaluated on a system equipped with an Intel Core i5 processor running at 3.20 GHz, 8 GB of RAM, and the Windows 10 Pro operating system. The implementation is developed using Python programming language. Visual Studio Code and Jupyter Notebook with IPYNB support are used for coding and experimental execution.

J. Glob. Res. Multidiscip. Studies, 2 (11):15-22
DOI: https://doi.org/xxxxxxxxxxxx

B. *Classification Performance Analysis*

The proposed model is trained on the ToN-IoT dataset and evaluated using accuracy, precision, recall, and F1-score, as presented in Table II. The performance matrix compares the proposed Convolutional Neural Network (CNN) and Random Forest (RF) models for network intrusion detection. Both models achieved 99.99% accuracy, demonstrating excellent classification performance. RF obtained 99.98% precision, 99.99% recall, and a 99.99% F1-score, while CNN achieved 99.97% precision, 99.98% recall, and a 99.97% F1-score. Overall, RF demonstrated slightly better performance across the evaluated metrics, indicating its effectiveness in detecting network intrusions.

TABLE II. CLASSIFICTION RESULTS OF PROPOSED MACHINE LEARNING AND DEEP LEARNING MODEL FOR NETWORK INTRUSION DETECTION FRAMEWORK

Matrix	Convolutional Neural Network (CNN)	Random Forest (RF)
Accuracy	99.99	99.99
Precision	99.97	99.98
Recall	99.98	99.99
F1-score	99.97	99.99

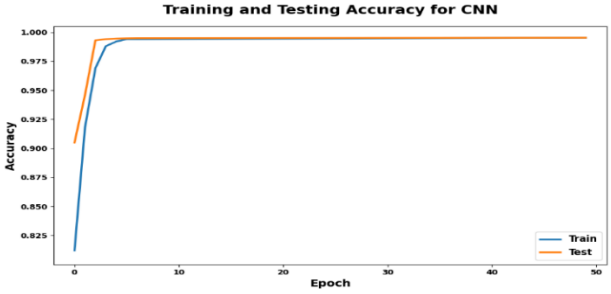

Fig. 6. Training and Testing Accuracy vs. Epoch for the Proposed Convolutional Neural Network (CNN) Model

Figure 6 illustrates the training and testing accuracy of the proposed Convolutional Neural Network (CNN) model across 50 epochs. Both curves increase rapidly during the initial epochs and gradually stabilize as training progresses. The close alignment of the training and testing curves indicates consistent learning performance and good generalization across both datasets.

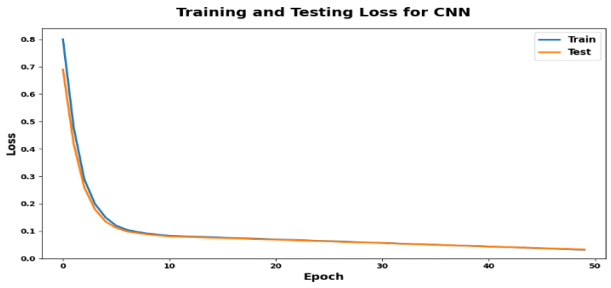

Fig. 7. Training and Testing Loss vs. Epoch for the Proposed Convolutional Neural Network (CNN) Model

Figure 7 illustrates the training and testing loss of the proposed Convolutional Neural Network (CNN) model across 50 epochs. Both curves decrease rapidly during the initial training phase and gradually decline as the epochs progress. The curves eventually converge toward a lower loss level, indicating that the model is learning effectively and minimizing prediction errors. The close alignment of the training and testing loss curves suggests stable learning behavior and good generalization performance.

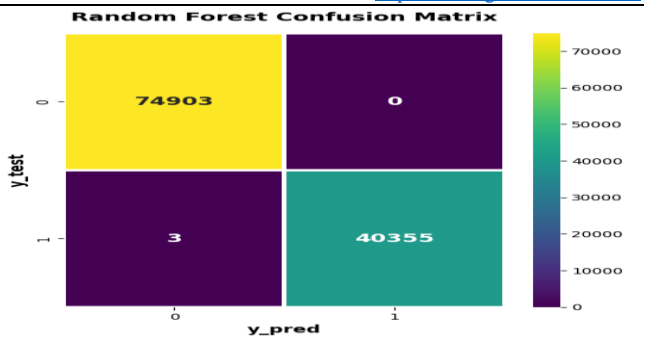

Fig. 8. Confusion Matrix of the Proposed Random Forest (RF) Model

Figure 8 presents the confusion matrix illustrates the classification performance of the proposed Random Forest (RF) model by comparing the predicted labels (y_pred) with the actual labels (y_test) across 115,261 test samples. The model correctly classified 74,903 true negative instances (class 0) and 40,355 true positive instances (class 1), with zero false positives and only 3 false negatives. These results demonstrate the model's excellent classification performance, high predictive reliability, and strong ability to distinguish between the two classes.

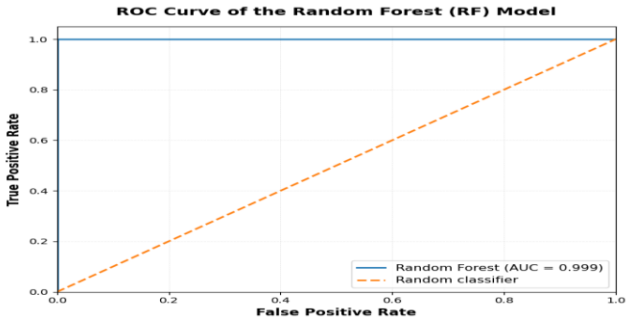

Fig. 9. Receiver Operating Characteristic (ROC) Curve of the Proposed Random Forest (RF) Model

Figure 9 The ROC curve illustrates the classification performance of the proposed Random Forest (RF) model, achieving an Area Under the Curve (AUC) of 99.9%. The curve approaches the upper-left corner, indicating a high true positive rate and a low false positive rate. Compared with the diagonal reference line representing random classification, the ROC curve demonstrates excellent discriminatory capability and strong performance in distinguishing between positive and negative classes.

C. *Comparative Analysis*

To assess the effectiveness of the proposed CNN and RF models, a comparative performance evaluation against existing models is presented in Table III. The evaluation includes Linear Discriminant Analysis (LDA), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Extreme Gradient Boosting (XGBoost), Gradient Boosting Machine (GBM), the proposed Convolutional Neural Network (CNN), and Random Forest (RF). The accuracy values 77% for LDA, 81% for SVM, 95.1% for KNN, 96.50% for XGBoost, and 97.17% for GBM. In comparison, the proposed CNN and RF models achieved 99.99% accuracy, demonstrating superior classification performance for anomaly detection. Among the evaluated models, the proposed approaches show the strongest accuracy results for identifying anomalous network activities.

TABLE III. PERFORMANCE COMPARISON OF MODELS FOR AUTOENCODER-BASED ANOMALY DETECTION

Model	Accuracy	Precision	Recall	F1-score
LDA [27]	0.77	0.79	0.77	0.77
SVM[28]	81	72	76	74
KNN[29]	95.1	93.5	96.9	95.2
XGBoost[30]	96.50	88.32	84.94	81.23
GBM[31]	97.17	97	97	97
Proposed CNN	99.99	99.97	99.98	99.97
Proposed RF	99.99	99.98	99.99	99.99

D. Discussion and Analysis

The proposed CNN and RF models achieved an accuracy of 99.99%, demonstrating excellent performance in network anomaly detection. The CNN model effectively learns complex traffic patterns, supporting the identification of abnormal network activities. The RF model provides robust classification performance with high precision and recall. Overall, both models offer reliable detection capabilities for strengthening network security against potential cyber threats.

V. CONCLUSION AND FUTURE STUDY

The increasing complexity and frequency of cybersecurity threats have created significant challenges for traditional intrusion detection and network monitoring systems. Conventional detection methods often struggle to identify evolving cyberattacks and previously unseen malicious activities due to their dependence on predefined patterns. This research presents a Network Anomaly Detection Framework for Unknown Cyberattacks using the TON_IoT dataset, incorporating data preprocessing, SMOTE-ENN balancing, and Select Best feature selection. The proposed CNN and RF models achieved an accuracy of 99.99%, outperforming LDA (77%), SVM (81%), KNN (95.1%), XGBoost (96.50%), and GBM (97.17%). These results demonstrate the potential of machine learning and deep learning techniques to improve network anomaly detection and strengthen cybersecurity against emerging threats.

The limitation of this study is that the proposed CNN and RF models are evaluated using the TON_IoT dataset, which may not fully represent all real-world network environments and emerging cyberattacks. In future, the framework will be validated with different datasets, optimized for network model efficiency, and real-time detection of zero-day attacks will be a focus.

- **Funding:** This research received no external funding.
- **Institutional Review Board Statement:** Not applicable.
- **Informed Consent Statement:** Not applicable.
- **Data Availability Statement:** The data supporting the findings of this study are available from the corresponding author upon reasonable request.
- **Conflicts of Interest:** The author declares no conflict of interest.

How to cite this article:
B. K. Sharma, "An Autoencoder-Based Network Anomaly Detection Framework for Unknown Cyberattacks," *Journal of Global Research in Multidisciplinary Studies,* vol. 2, no. 11, pp. 15–22, Nov. 2026, Doi: XXXX

REFERENCES

[1] K. Korniszuk and B. Sawicki, "Autoencoder-Based Anomaly Detection in Network Traffic," in *2024 25th International Conference on Computational Problems of Electrical Engineering (CPEE)*, IEEE, Sep. 2024, pp. 1–4. doi: 10.1109/CPEE64152.2024.10720411.

[2] A. B. Owolabi, "A Multi-Layer Deep Learning Framework for Intelligent Cyber Attack Prevention," *Int. J. Res. Innov. Appl. Sci.*, vol. 11, no. 6, pp. 2521–2541, 2026, doi: 10.51584/IJRIAS.2026.11060195.

[3] Q. P. Nguyen, K. W. Lim, D. M. Divakaran, K. H. Low, and M. C. Chan, "GEE: A Gradient-based Explainable Variational Autoencoder for Network Anomaly Detection," in *2019 IEEE Conference on Communications and Network Security (CNS)*, IEEE, Jun. 2019, pp. 91–99. doi: 10.1109/CNS.2019.8802833.

[4] C. Park, J. Lee, Y. Kim, J.-G. Park, H. Kim, and D. Hong, "An Enhanced AI-Based Network Intrusion Detection System Using Generative Adversarial Networks," *IEEE Internet Things J.*, vol. 10, no. 3, pp. 2330–2345, Feb. 2023, doi: 10.1109/JIOT.2022.3211346.

[5] T. A. Khan *et al.*, "Multi-Source Cyber Intrusion Detection Using Ensemble Machine Learning," *J. Comput. Sci.*, vol. 21, no. 1, pp. 111–123, Jan. 2025, doi: 10.3844/jcssp.2025.111.123.

[6] G. Prabhakar and B. B. Rao, "Enhanced Deep Learning-Based Security Model for Data in Cloud," *Int. J. Electron. Commun. Eng.*, vol. 12, no. 3, pp. 88–100, Mar. 2025, doi: 10.14445/23488549/IJECE-V12I3P108.

[7] F. W. Alsaade and M. H. Al-Adhaileh, "Cyber Attack Detection for Self-Driving Vehicle Networks Using Deep Autoencoder Algorithms," *Sensors*, vol. 23, no. 8, p. 4086, Apr. 2023, doi: 10.3390/s23084086.

[8] Y. Song, S. Hyun, and Y.-G. Cheong, "Analysis of Autoencoders for Network Intrusion Detection," *Sensors*, vol. 21, no. 13, p. 4294, Jun. 2021, doi: 10.3390/s21134294.

[9] K. Thirumalaisamy, "Zero-Day Vulnerabilities in Container Images: Risks and Detection using Generative AI," *Int. J. Sci. Res.*, vol. 14, no. 6, pp. 309–317, Jun. 2025, doi: 10.21275/SR25603123105.

[10] P. G. Himanshu Kumar *et al.*, "The Journey to Intent-based Networking: Ten Key Principles for Accelerating Adoption," *IEEE Access*, 2017.

[11] Nisha, "Machine learning-based intrusion and anomaly detection for enhancing security in IoT networks using BoT-IoT dataset," *Int. J. Eng. Comput. Sci.*, vol. 6, no. 2, pp. 241–248, Jul. 2024, doi: 10.33545/26633582.2024.v6.i2c.174.

[12] B. A. Manjunatha, K. A. Shastry, E. Naresh, P. K. Pareek, and K. T. Reddy, "A network intrusion detection framework on sparse deep denoising auto-encoder for dimensionality reduction," *Soft Comput.*, vol. 28, no. 5, pp. 4503–4517, Mar. 2024, doi: 10.1007/s00500-023-09408-x.

[13] F. S. Alrayes, M. Zakariah, S. U. Amin, Z. Iqbal Khan, and M. Helal, "Intrusion Detection in IoT Systems Using Denoising Autoencoder," *IEEE Access*, vol. 12, pp. 122401–122425, 2024, doi: 10.1109/ACCESS.2024.3451726.

[14] D. K. Jayswal and S. Gupta, "Intelligent Detection of Anomalous Online Payments Using Machine Learning for Financial Cyber Defense," in *2026 International Conference on Intelligent Digitization of Systems and Services (IDSS)*, San Antonio, TX, USA: IEEE, May 2026, pp. 102–108. doi: 10.1109/IDSS69894.2026.11691329.

[15] K. Siddique, Z. Akhtar, F. Aslam Khan, and Y. Kim, "KDD Cup 99 Data Sets: A Perspective on the Role of Data Sets in Network Intrusion Detection Research," *Computer (Long. Beach. Calif).*, vol. 52, no. 2, pp. 41–51, Feb. 2019, doi: 10.1109/MC.2018.2888764.

[16] R. Chitrakar and C. Huang, "Selection of Candidate Support Vectors in incremental SVM for network intrusion detection," *Comput. Secur.*, vol. 45, pp. 231–241, Sep. 2014, doi: 10.1016/j.cose.2014.06.006.

[17] F. Harrou, B. Bouyeddou, A. Dairi, and Y. Sun, "Exploiting Autoencoder-Based Anomaly Detection to Enhance Cybersecurity in Power Grids," *Futur. Internet*, vol. 16, no. 6, p. 184, May 2024, doi: 10.3390/fi16060184.

[18] A. Y. D. Siale, Q. M. Z. Hassan, M. F. A. S. Kadekle, and B. S. Veena, "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, May 2025, doi: 10.18196/jrc.v6i3.25169.

[19] M. H. Hamza Afzal, "Securing AI Systems Against Adversarial

Attacks: A Framework for Building Robust and Trustworthy Machine Learning Models," *Comput. Fraud Secur.*, no. 5, pp. 65–74, May 2024, doi: 10.52710/cfs.867.

[20] A. Bhushan, N. Kaur, Monisha, M. Srivastava, S. Sethi, and M. Kaur, "AI-Powered Neural Network System for Real-Time Network Anomaly Detection in Cybersecurity," in *2026 5th International Conference on Innovative Practices in Technology and Management (ICIPTM)*, IEEE, Feb. 2026, pp. 1–6. doi: 10.1109/ICIPTM69057.2026.11465896.

[21] M. Bachar, A. Khiat, and K. El Guemmat, "A Multi-Stage Hybrid Anomaly Detection Framework for IoT Security Using Machine Learning and Deep Learning," in *2026 International Conference on Circuit, Systems and Communication (ICCSC)*, IEEE, Jul. 2026, pp. 1–6. doi: 10.1109/ICCSC71566.2026.11650024.

[22] A. J. M. Rani, A. Rajalingam, O. L, T. S. Rao, R. Shalinirajan, and V. S. Pandi, "An Innovative Use of Deep Learning in Cybersecurity: Implement Sequential Pattern Recognition and Autoencoders for Real-Time Intrusion Detection," in *2025 International Conference on Future Technologies (ICFT)*, IEEE, Nov. 2025, pp. 1–7. doi: 10.1109/ICFT66708.2025.11336606.

[23] S. K. Udayakumar, H. Ragothaman, and K. M. Khare, "A Novel Dataset and a Hybrid Ensemble Approach for Anomaly Detection in Enterprise-Access-Logs for Anomaly Detection," in *2025 International Conference on Data Science, Agents & Artificial Intelligence (ICDSAAI)*, IEEE, Mar. 2025, pp. 1–6. doi: 10.1109/ICDSAAI65575.2025.11011761.

[24] N. P. A and S. Ravindran, "Cloud-Based Adaptive Detection Framework for IIoT Networks Using Machine Learning Techniques," in *2025 First International Conference on Intelligent Computing and Systems at the Edge (ICEdge)*, IEEE, Dec. 2025, pp. 1–5. doi: 10.1109/ICEdge67252.2025.11412464.

[25] M. B. Vemula, K. P. Kumar, H. R. S. Thipparthi, and P. S. Jasti, "Network Anomaly Detection Using Generative Adversarial Networks," in *2024 First International Conference on Software, Systems and Information Technology (SSITCON)*, IEEE, Oct. 2024, pp. 1–6. doi: 10.1109/SSITCON62437.2024.10796515.

[26] A. Almeida, L. Gonçalves, C. Zanchettin, and B. L. D. Bezerra, "A Semi-Supervised Autoencoder Approach for Efficient Intrusion Detection in Network Traffic," in *2023 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, IEEE, Oct. 2023, pp. 1–6. doi: 10.1109/LA-CCI58595.2023.10409491.

[27] N. W. Khan *et al.*, "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.

[28] M. S. Abdullah Sami and M. Abid, "Unsupervised Anomaly Detection for Smart IoT Devices: Performance and Resource Comparison," in *2025 IEEE 2nd International Conference on Computing, Applications and Systems (COMPAS)*, IEEE, Oct. 2025, pp. 1–5. doi: 10.1109/COMPAS67506.2025.11381608.

[29] A. M. Almasabi, M. Khemakhem, F. E. Eassa, A. Ahmed Abi Sen, A. B. Alkhodre, and A. Harbaoui, "A Smart Framework to Detect Threats and Protect Data of IoT Based on Machine Learning," *IEEE Access*, vol. 12, no. November, pp. 176833–176844, 2024, doi: 10.1109/ACCESS.2024.3498603.

[30] H. Alamro *et al.*, "An intelligent deep representation learning with enhanced feature selection approach for cyberattack detection in internet of things enabled cloud environment," *Sci. Rep.*, vol. 15, no. 1, p. 34013, Sep. 2025, doi: 10.1038/s41598-025-13457-7.

[31] N. Dharini, V. S. Janani, and J. Katiravan, "Efficient detection of intrusions in TON-IoT dataset using hybrid feature selection approach," *Sci. Rep.*, vol. 16, no. 1, p. 7763, Feb. 2026, doi: 10.1038/s41598-026-37834-y.