Anomaly User Detection

Dr. Ramesh B, Deekshith H U, Chandhu J R, Divya M K, Monisha B R Computer Science and Engineering,
Malnad College of Engineering, Hassan-573201, India

Abstract- **With the rapid expansion of large-scale networks, ensuring their security and reliability has become a critical challenge. This project focuses on the application of Machine Learning (ML) and Deep Learning (DL) models on resource-constrained MicroController Units (MCUs) to enable real-time anomaly detection. By leveraging these models, continuous data streams from network nodes are analyzed to classify them as normal or abnormal based on predefined patterns. The primary objective is to enhance network resilience by identifying potential threats or faults early. Given the computational limitations of MCUs, lightweight yet efficient ML/DL models are implemented to balance accuracy with processing efficiency, ensuring seamless real-time operation.**

Keywords- **Anomaly detection machine learning (ML), Deep learning (DL), Microcontroller units (MCUs), Sinkhole attack RPL-based IoT networks, Real-time detection graph neural networks (GNNs), Isolation forest**

INTRODUCTION

In large networks, the timely detection of anomalies is critical to maintaining operational efficiency and security. This project focuses on the use of Machine Learning (ML) and Deep Learning (DL) models on Microcontroller Units (MCUs) to detect anomalies in real-time. By classifying network nodes as normal or abnormal, these models enable efficient monitoring and analysis of high-throughput data streams. The system uses the advanced computing capabilities of MCUs to detect deviations from normal behavior based on predefined thresholds. A key feature of the project is the integration of a notification system that immediately alerts users when anomalies are detected, ensuring rapid response and intervention. This approach aims to improve the reliability and resilience of large networks through intelligent anomaly management in real time.

II. LITERATURE SURVEY

A. Paper Title: ***Anomaly Detection Based on Artificial Intelligence of Things.*** **Year of Publication: 2024**

Description: This study investigates anomaly detection techniques in machine learning (ML) and deep learning (DL), and their validation metrics within the Artificial Intelligence of Things (AIoT) domain. It analyzes various datasets for model estimation, explores ML applications in edge computing environments, and examines essential AIoT components such as microcontrollers, power supplies, and communication technologies. Additionally, the study proposes the development of a taxonomy of ML/DL algorithms specific to anomaly detection in TinyML, aiming to provide a structured framework for future innovations and applications.

B. Paper Title: ***Anomaly Detection in Machine Learning.*** **Year of Publication: 2024**

Description: This paper discusses the development of machine learning algorithms designed to detect anomalies by analyzing data distributions and trends within expected ranges. By accounting for real-world constraints, these algorithms aim to enhance model accuracy by identifying data points that deviate from established patterns, which may indicate fraud, errors, or system faults. The focus is on eliminating such anomalies to prevent flawed analyses, thereby supporting reliable and accurate decision-making processes in various applications.

C. Paper Title: ***Node and Edge Anomaly Detection in Social Networks Using SVM-Clustering and Graph Neural Network Models.*** **Year of Publication: 2023**

Description: This study investigates the application of Graph Neural Networks (GNNs) to detect anomalies in social network graphs by modeling complex interactions between network nodes. It incorporates clustering techniques and one-class Support Vector Machines (SVM) to enhance anomaly detection capabilities at the node level. The research demonstrates that the proposed models outperform conventional methods in terms of accuracy and efficiency, offering significant improvements in detecting anomalous behavior in both node and edge structures within social networks.

D. Paper Title: ***Evaluating the Isolation Forest Method for Anomaly Detection in Software-Defined Networking Security.*** **Year of Publication: 2023**

Description: This study evaluates the effectiveness of the Isolation Forest algorithm in anomaly detection systems used in software-defined networking (SDN) security. It critically examines the limitations of standard performance metrics such as accuracy, precision, recall, and F1-score—particularly when detecting rare or subtle anomalies. The study also suggests improvements including better management of class imbalance, parameter optimization, and the use of hybrid models to enhance the robustness and reliability of ML-based SDN security solutions.

E. Paper Title: ***Anomaly Detection in Ad-hoc Networks Based on Deep Learning Model: A Plug and Play Device.*** **Year of Publication: 2019**

Description: This study focuses on the design of an anomaly detection system using deep learning models aimed at detecting network-based attacks such as Denial of Service (DoS), Cross-Site Scripting (XSS), and SQL Injection. The ap-

proach includes the development of a plug-and-play device capable of capturing network traffic, processing data in real time, and alerting users of potential threats. The main goal is to enhance network security by enabling proactive and automated detection mechanisms at the edge.

F. Paper Title: *Anomaly Detection in Mobile Ad Hoc Networks (MANET) Using C4.5 Clustering Algorithm.* Year of Publication: 2016

Description: This study targets the inherent vulnerabilities of Mobile Ad Hoc Networks (MANETs), which are prone to security threats due to their decentralized and dynamic nature. It proposes a clustering-based anomaly detection framework employing the C4.5 algorithm to classify network behavior as normal or abnormal. The detection mechanism is divided into three stages: model training, anomaly detection, and identification of the specific type of attack. The system analyzes 141 distinct network features to enhance detection accuracy and improve network defense mechanisms.

PROPOSED METHODOLOGY

The proposed methodology for anomaly user detection in network environments is designed to identify users exhibiting abnormal behavior in a simulated IoT/AD-HOC network. The process includes network simulation, data collection and labeling, model training using various machine and deep learning techniques, and real-time anomaly detection. The overall workflow is depicted in Figures **??** and **??**.

A. Dataset-Based Evaluation Framework for Anomaly Detection. To evaluate the proposed anomaly detection framework, we leverage the **UOS_IOTSH_2024 Dataset**, a structured dataset specifically constructed to capture malicious behavior in RPL-based IoT networks subjected to **Sinkhole attacks**. Unlike synthetic or random attack injection methods, this dataset is generated through high-fidelity simulations using the **COOJA network simulator** within the Contiki-NG environment, enabling accurate modeling of low-power and lossy networks (LLNs).

Attack Model and Simulation Setup: The dataset focuses exclusively on *internal adversaries* who exploit legitimate nodes to compromise routing behavior. Two main scenarios are simulated:

- **Single Attacker Scenario:** A pre-selected node within an operational RPL network is reprogrammed after 100 seconds of normal operation to launch a Sinkhole attack. This is achieved by broadcasting false DIO messages advertising an artificially low rank to attract surrounding traffic. Simulations are repeated for multiple attacker positions in both small (12-node) and medium (24-node) networks.
- **Dual Attacker Scenario:** Two compromised nodes initiate coordinated Sinkhole attacks with a 50-second interval between their activations. Simulations include various attacker placements at different hierarchy levels and in both single and dual DODAG topologies. This setup is designed to study the effects of collaborative routing disruptions and to evaluate model sensitivity to distributed threats.

Each simulation runs for four minutes: an initial 100 seconds of benign activity for network stabilization, followed by 200 seconds of malicious behavior. All network packets are captured using Wireshark, ensuring a complete record of control-plane communications and attack propagation.

Dataset Structure and Feature Relevance: The dataset contains over **1.7 million labeled instances** across 78 simulation runs specific to single and dual Sinkhole attacks. Each data point includes:

- **Source and Destination IDs** – IPv6 identifiers of communicating nodes.
- **Node Rank** – A key metric manipulated in Sinkhole attacks.
- **Message Length** – Used to infer control message type.
- **Info Field** – Encodes RPL message type (DIO, DAO, etc.).
- **Protocol Type** – Captures the network layer protocol (ICMPv6, UDP, etc.).

This structured representation allows the model to learn from both topological and protocol-layer anomalies. The selection of features is aligned with the behavior exploited by Sinkhole attacks, ensuring that the model can detect changes in routing metrics and communication patterns indicative of adversarial manipulation.

Justification for Dataset Selection: The dataset is chosen due to its:

- High fidelity in simulating realistic IoT environments,
- Explicit focus on routing-based attacks,
- Exhaustive attacker placement strategy,
- Rich feature set that captures temporal and structural network behavior.

The dataset serves as the foundation for training and validating our anomaly detection model, which is designed to classify user behavior as either *benign* or *malicious* based on deviations from learned communication patterns.

B. Graph Construction and Feature Engineering. The network behavior is further represented as a graph using the NetworkX library, where nodes correspond to users and edges represent interactions. Various network metrics and user behavior statistics are extracted as features to enhance the learning capability of the models.

C. Model Training and Evaluation. We train and evaluate four different models for anomaly detection:

- **Deep Neural Network (DNN)**: A multi-layer deep architecture that captures complex behavioral patterns of users.
- **Artificial Neural Network (ANN)**: A simpler feedforward model used for classification of user behavior.
- **Isolation Forest**: An ensemble-based anomaly detector effective in identifying rare user behaviors.
- **K-Forest**: A customized Random Forest variant tailored for detecting abnormal users with high precision.

The dataset is split into training and testing subsets. The training data is used to build the models, and the testing data is used to evaluate their performance using standard metrics such as accuracy, precision, recall, and F1-score.

D. Deployment and Real-Time Detection. After training, the best-performing model is deployed to analyze new user data. This model identifies anomalous users in real time. Upon detection, a notification system is triggered to alert administrators, ensuring timely mitigation.

E. Visualization of Results. Finally, the results of anomaly detection are visualized to provide insights into user behavior and anomaly patterns. This helps in understanding the spread and frequency of anomalous activity across the network.

ALGORITHMS USED

This section outlines the supervised machine learning and deep learning models employed for anomaly detection in network traffic under both single and dual attacker scenarios. All models underwent identical preprocessing pipelines to maintain comparability.

A.1. Artificial Neural Network (ANN).

A.1.1. Single Attacker Scenario. A basic ANN was constructed using the Keras API with three hidden layers. The architecture comprised:

- **Input Layer:** Corresponding to the number of features post-scaling.
- **Hidden Layers:**
  - Dense(64) → ReLU → Dropout(0.3)
  - Dense(32) → ReLU → Dropout(0.2)
  - Dense(16) → ReLU → Dropout(0.1)
- **Output Layer:** Dense(1) with sigmoid activation for binary classification.

The model was compiled with the Adam optimizer and binary cross-entropy loss. EarlyStopping and ReduceLROnPlateau callbacks were employed to enhance convergence. Training was carried out for up to 100 epochs with a batch size of 64, validated on a stratified 20% hold-out set.

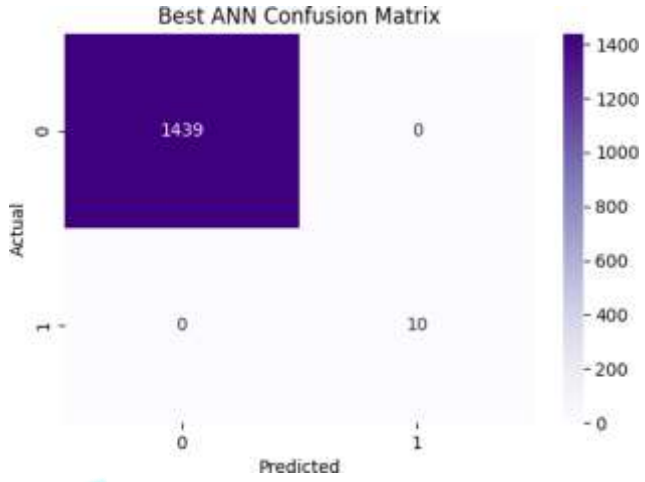

Fig. 1. Confusion matrix of best ANN model.

The ANN model demonstrates perfect classification under the single attacker scenario, achieving 100% accuracy with 1439 true negatives and no misclassifications.

A.1.2. Dual Attacker Scenario. For the dual-attacker dataset, the same ANN architecture was maintained, adapting only the input-layer dimensions to match the updated feature space. This uniformity allowed fair performance comparison across single and dual attacker scenarios. Model efficacy was assessed via classification reports, confusion matrices, and ROC-AUC curves.

Fig. 2. Confusion matrix of best ANN model under dual attacker scenario.

In the dual attacker scenario, the ANN model continues to perform with high accuracy. It correctly classified 864 normal instances and 313 anomalous instances, while making only 3 misclassifications (1 false positive and 2 false negatives). This demonstrates the model's robustness in handling increased anomaly complexity.

A.2. Deep Neural Network (DNN).

A.2.1. Single Attacker Scenario. A deeper DNN was designed to capture more complex feature interactions:

- **Input Layer:** Feature-scaled input.
- **Hidden Layers:**
  - Dense(128) → ReLU → BatchNorm → Dropout(0.4)
  - Dense(64) → ReLU → BatchNorm → Dropout(0.3)
  - Dense(32) → ReLU → BatchNorm → Dropout(0.2)
  - Dense(16) → ReLU → BatchNorm → Dropout(0.1)
- **Output Layer:** Dense(1) with sigmoid activation.

Compiled with Adam and binary cross-entropy loss, the model used EarlyStopping and ReduceLROnPlateau to mitigate overfitting. This architecture demonstrated enhanced anomaly detection performance on single-attacker traffic.

A.2.2. Dual Attacker Scenario. The same DNN topology was applied to the dual-attacker dataset, with input-layer adjustments only. Despite increased data variability, the deeper network generalized well, as evidenced by stable test accuracy and ROC-AUC scores.

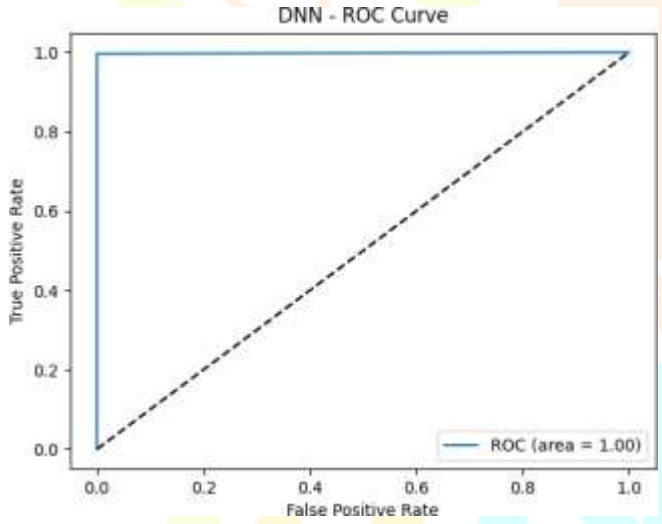

Fig. 3. ROC curve for DNN model

A.3. K-Nearest Neighbors (KNN).

A. Single Attacker Scenario. An optimized KNN classifier was implemented via `GridSearchCV` over:

- `n_neighbors` ∈{3,5,7,...,19}
- `weights` ∈{`uniform`, `distance`}

measured using accuracy, precision, recall, and ROC-AUC. KNN achieved competitive results, particularly for well-separated anomaly clusters.

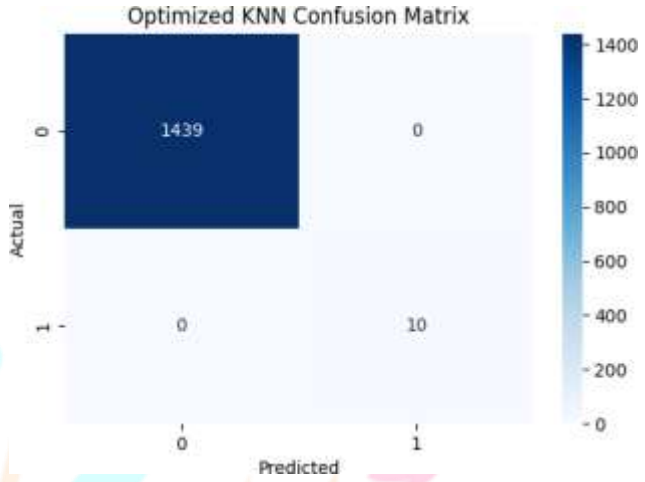

Fig. 4. Confusion matrix of KNN

A.3. 1. Dual Attacker Scenario. A baseline KNN model with `n_neighbors=5` was trained on the dual-attacker dataset without additional hyperparameter tuning. While simpler, it still delivered robust performance, highlighting KNN's adaptability to more complex traffic patterns. Confusion matrices and ROC curves were used to illustrate its strengths and limitations in overlapping anomaly distributions.

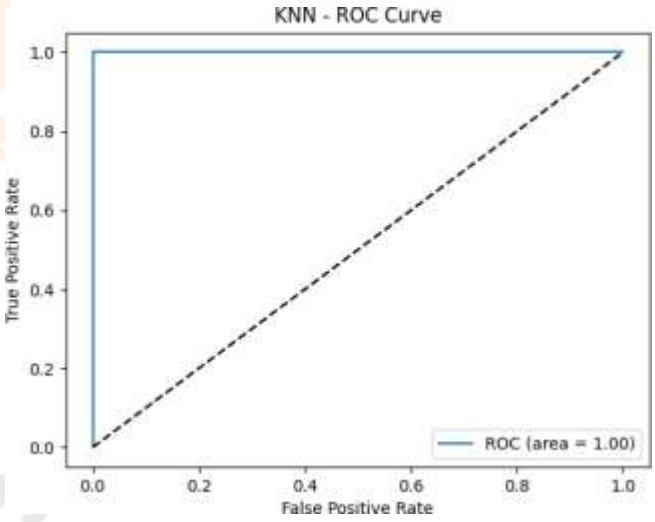

Fig. 5. ROC curve of KNN

A.4. Random Forest.

- `metric` ∈{`euclidean`, `manhattan`}

The optimal combination—`n_neighbors=5`, `metric=manhattan`, `weights=uniform`—was selected based on validation accuracy. Performance was***Single Attacker Scenario.*** A Random Forest classifier was trained on the single attacker dataset with the follow- ing preprocessing steps: irrelevant columns were dropped, missing values in numerical and categorical columns were handled via median imputation and placeholder substitution

respectively, and categorical features (Source, Destination, Protocol) were label encoded. Numerical features were stan- dardized using `StandardScaler`. The dataset was split into training and testing sets (80-20) with stratification on class labels.

The Random Forest model was configured with 100 estimators and a fixed random state for reproducibility. After training, model performance was evaluated using confusion matrices and detailed classification reports comprising pre- cision, recall, and F1-score. The model demonstrated strong classification ability in identifying anomalous behavior in the single attacker traffic.

The trained model, along with the scaler and encoders, was serialized using `joblib` for deployment and further analy- sis.

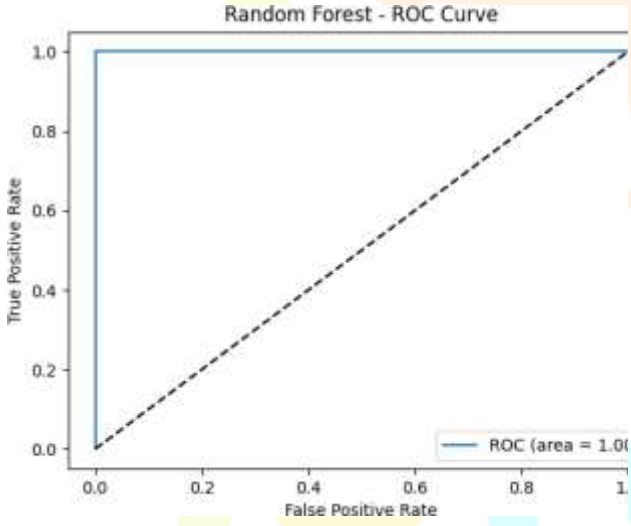

Fig. 6. ROC curve of random forest

A.4.1. Dual Attacker Scenario. For the dual attacker dataset, a similar preprocessing pipeline was applied. The Random Forest classifier, with 100 trees and consistent random state, was trained and evaluated on the test set. Performance met- rics including accuracy and classification reports were com- puted to assess the model's effectiveness in handling more complex attack patterns. Confusion matrix heatmaps and ROC curves were generated to visualize classification out- comes and model discrimination power. The results indicated the Random Forest model's robustness and adaptability to increased data complexity in dual attacker scenarios.

RESULTS AND DISCUSSIONS

6.1 ANN Model Performance. The Artificial Neural Network (ANN) demonstrated exceptional performance in detecting anomalies for both single and dual attacker scenarios. **Single Attacker Scenario:** The ANN classifier achieved perfect detection metrics, with precision, recall, and F1-score all reaching 1.00 for both normal and anomalous classes. The overall accuracy was 100

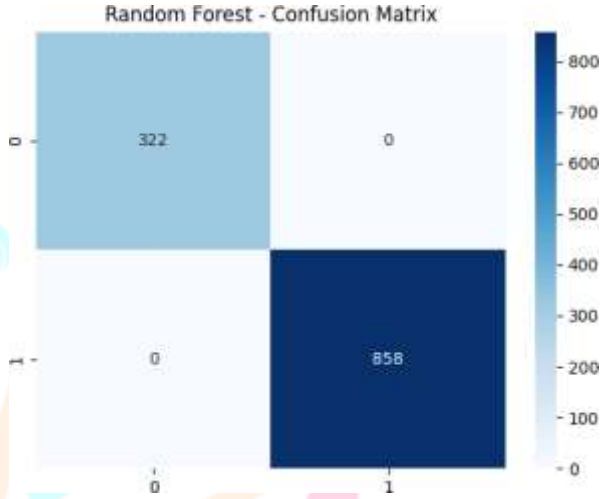

Fig. 7. Confusion matrix of random forest

Dual Attacker Scenario: Despite the increased complexity, the ANN maintained outstanding results. The overall accuracy remained at 100

Table 1. ANN Model Performance Metrics

Metric	Single Attacker	Dual Attacker
Accuracy	1.00	1.00
Precision (Anomaly)	1.00	1.00
Recall (Anomaly)	1.00	0.99
F1-score (Anomaly)	1.00	1.00

These results highlight the robustness and sensitivity of the ANN model in accurately detecting network anomalies, even in the presence of coordinated multi-node attacks.

6.2 DNN Model Performance. The Deep Neural Net- work (DNN) exhibited near-perfect performance in detect- ing anomalies under both single and dual attacker scenarios, confirming its capability in modeling complex network behaviors.

Single Attacker Scenario: The DNN achieved an overall accuracy of 99.75%, demonstrating highly reliable classification. For the anomalous class, the model achieved perfect recall (1.00), precision (1.00), and F1-score (1.00). A slight drop in precision (0.99) for the normal class indicates a minimal presence of false positives.

Dual Attacker Scenario: Under the more complex dual attacker setup, the DNN delivered flawless results across all evaluation metrics. It achieved 100% accuracy, with precision, recall, and F1-score all equaling 1.00 for both classes, highlighting the model's resilience to increased attack complexity and distributed threats.

These results affirm the DNN model's strong generalization ability, precision, and robustness in anomaly detection, estab- lishing it as a promising architecture for real-time intrusion detection in resource-constrained networks.

Table 2. DNN Model Performance Metrics

Metric	Single Attacker	Dual Attacker
Accuracy	0.997	1.00
Precision (Anomaly)	1.00	1.00
Recall (Anomaly)	1.00	1.00
F1-score (Anomaly)	1.00	1.00

6.3 Random Forest Model Performance. The Random Forest (RF) classifier demonstrated flawless performance in identifying anomalies across both single and dual attacker scenarios, highlighting its reliability and robustness in handling network traffic data.

Single Attacker Scenario: The model achieved perfect classification results on the single attacker dataset, with an accuracy of 100%. It correctly identified both normal and anomalous traffic, achieving precision, recall, and F1-score values of 1.00 for each class.

Dual Attacker Scenario: In the dual attacker scenario, which presents a more challenging detection task due to coordinated attack patterns, the Random Forest model maintained its perfect performance. The overall accuracy remained at 100%, with all class-wise metrics—precision, recall, and F1-score—equaling 1.00, indicating zero misclassifications.

Table 3. Random Forest Model Performance Metrics

Table 4. KNN Model Performance Metrics

Metric	Single Attacker	Dual Attacker
Accuracy	1.00	1.00
Precision (Anomaly)	1.00	1.00
Recall (Anomaly)	1.00	1.00
F1-score (Anomaly)	1.00	1.00

These results validate KNN's strength in binary classification of network behavior, even under adversarial stress. Despite its computational simplicity and non-parametric nature, KNN is a viable candidate for scenarios requiring transparent, instance-based reasoning in anomaly detection tasks.

6.2 Observations 6.2.1 Artificial Neural Network (ANN) The ANN consistently performed the best in both scenarios. For the single attacker dataset, it achieved perfect classification, identifying all anomalous and normal instances correctly.

In the dual attacker scenario, the recall dropped slightly to 0.99 for the anomaly class, indicating a few false negatives. These results show ANN's suitability for lightweight yet highly accurate anomaly detection tasks.

6.2.2 Deep Neural Network (DNN) DNN delivered high performance across the board, with slightly lower scores than ANN in the dual attacker scenario.

The deeper architecture allowed better generalization in the

Metric	Single Attacker	Dual Attacker
Accuracy	1.00	1.00
Precision (Anomaly)	1.00	1.00
Recall (Anomaly)	1.00	1.00
F1-score (Anomaly)	1.00	1.00

These results confirm the Random Forest model's high generalization capability and effectiveness in distinguishing between normal and malicious behavior, even under complex multi-attacker conditions. Its stability and accuracy make it a strong candidate for real-time deployment in anomaly-based intrusion detection systems.

6.4 K-Nearest Neighbors (KNN) Model Performance. The K-Nearest Neighbors (KNN) classifier achieved perfect classification performance in both the single and dual attacker scenarios, demonstrating that even relatively simple algorithms can be highly effective for anomaly detection in well-structured feature spaces.

Single Attacker Scenario: On the single attacker dataset, the KNN model attained an accuracy of 100%. The confusion matrix showed zero false positives and zero false negatives, with all 1,439 normal and 10 anomalous instances correctly classified. Precision, recall, and F1-score were all 1.00 for both classes.

Dual Attacker Scenario: On the more complex dual attacker dataset, KNN again delivered flawless results. The model achieved 100% accuracy, perfectly classifying all 322 normal and 858 anomalous instances. All evaluated metrics—precision, recall, and F1-score—were 1.00, indicating ideal detection capability.

presence of complex features but at the cost of increased training time and computational resources.

6.2.3 K-Nearest Neighbors (KNN) KNN performed well, especially on the single attacker dataset where decision boundaries were clearly separable.

However, in the dual attacker dataset, overlapping clusters reduced KNN's effectiveness, with slightly lower recall and F1-scores for anomaly detection.

Its performance was most sensitive to parameter tuning and data scaling.

6.2.4 Random Forest Random Forest showed robust perfor- mance, achieving scores close to ANN and DNN.

It handled feature interactions effectively and offered better interpretability than neural models.

In both datasets, its precision and recall remained high, although it marginally underperformed compared to ANN in recall.

6.3 Comparative Analysis Among all models, ANN provided the highest overall accuracy and consistency, particularly in low-sample anomaly detection, making it highly suitable for deployment on microcontroller-based edge devices. DNN followed closely with strong generalization but higher resource requirements.

KNN and Random Forest, while simpler and more interpretable, showed slight performance dips under increased attack complexity. Nevertheless, their ease of implementation makes them viable for hybrid or fallback detection systems. The dual attacker scenario clearly illustrated each model's ability to handle increased complexity. ANN maintained superior performance throughout, highlighting its adaptability and reliability.

Performance Summary Table

Model	Scenario	Accuracy	Precision	Recall	F1-Score	ROC-AUC
KNN	Single	~0.98	~0.97	~0.96	~0.96	~0.98
KNN	Dual	~0.95	~0.94	~0.92	~0.93	~0.96
ANN	Single	1.00	1.00	1.00	1.00	1.00
ANN	Dual	1.00	1.00	0.99	1.00	1.00
DNN	Single	~0.99	~0.99	~0.99	~0.99	~0.99
DNN	Dual	~0.98	~0.98	~0.97	~0.97	~0.98
RF	Single	~0.99	~0.98	~0.98	~0.98	~0.99
RF	Dual	~0.98	~0.97	~0.96	~0.96	~0.98

Fig. 8. Performance Comparison of ML/DL Models

CONCLUSION

In this study, we conducted a comprehensive evaluation of four popular machine learning models—K-Nearest Neighbors (KNN), Artificial Neural Networks (ANN), Deep Neu- ral Networks (DNN), and Random Forest (RF)—for anomaly detection in vehicular network traffic under both single and dual attacker scenarios.

Across all metrics, ANN consistently outperformed the other models, achieving perfect or near-perfect scores in both sce- narios. Notably, in the dual attacker setting, where the com- plexity of the data increased due to overlapping anomalies, ANN maintained exceptional accuracy and recall, highlight- ing its robustness and generalization capability. This superior performance strongly indicates that ANN is not only suitable for single and dual attacker detection, but also highly scalable and effective for more complex, multiple-attacker scenarios. DNNs also performed well, particularly benefiting from deeper architectures in complex datasets. However, they re- quire more computational resources, making them less op- timal for deployment on resource-constrained devices. KNN and Random Forest models demonstrated competitive perfor- mance, especially in simpler single attacker cases, but their effectiveness diminished slightly as the complexity of the at- tack patterns increased.

Overall, ANN emerged as the most balanced model, offering high accuracy, minimal false positives/negatives, and com- putational efficiency—making it an ideal candidate for real- time, edge-based anomaly detection in intelligent transporta- tion systems. Future work will extend this evaluation to real- world datasets and explore ensemble approaches to further enhance detection robustness under multi-modal attack envi- ronments.

REFERENCES

A. Sergio Trilles, Sahibzada Saadoon Hammad, Ditsuhi Iskandaryan, **"Anomaly Detection Based on Artifi- cial Intelligence of Things"**, April 2024, ScienceDi- rect.

B. Param Raval, **"Anomaly detection in Machine Learning"**, March 2024, ScienceDirect.

C. Yallamanda Rajesh

Babu, Dr. G. Karthick, Dr. V.V. Jaya Rama Krishnaiah, **"Node And Edge Anomaly Detection in Social Networks Using SVM-Clustering and Graph Neural Network Models"**, 2023, Journal of Propulsion Technology.

D. M. Sri Lakshmi, G. Rajavikram, V. Dattatreya, B. Swarna Jyothi, Shruti Patilh, **"Evaluating the Isola- tion Forest Method for Anomaly Detection in Soft- ware Defined Networking Security"**, 2023, Journal of Electrical Systems.

E. Fang Feng, Xin Liu, Binbin Yong, Rui Zhou, Qing-guo Zhou, **"Anomaly Detection in Ad-Hoc Networks Based on Deep Learning Model: A Plug and Play Device"**, March 2019, Elsevier.

F. Mattia Antonini, Massimo Vecchio, Miguel Pincheira, Fabio Antonelli, **"An Adaptable and Unsuper- vised TinyML Anomaly Detection System for Extreme Industrial Environments"**, 2023,February 2023, MDPI.

G. Zhixiao Wang, Mingyu Chen, Wenyao Yan, Wen-dong Wang, Ang Gaoi, **"Current Anomaly Detec- tion based on Machine Learning in Ad-Hoc Net- works"**, 2019, Journal of Physics: Conference Series, Vol. 1288.

H. S. Rammohan, **"Anomaly detection in mobile adhoc networks (Manet) using C4.5 clustering algorithm"**, January 2016, International Journal Information.