National College of Ireland

Securing 5G IoT Networks: A Machine Learning Framework for Zero-Trust Intrusion Detection System

MSc Practicum 2
Master of Science in Cybersecurity

Preetham Charan Sridhar
Student ID: x23183683

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland

MSc Project Submission Sheet

Master of Science in Cybersecurity

Student Name:	PREETHAM CHARAN SRIDHAR
Student ID:	x23183683
Programme:	MSC Cyber Security **Year:** 2024-2025
Module:	MSc Practicum 2
Supervisor:	VIKAS SAHNI
Submission Due Date:	12-12-2024
Project Title:	Securing 5G IoT Networks: A Machine Learning Framework for Zero-Trust Intrusion Detection System
Word Count:	8018 **Page Count** 20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	PREETHAM CHARAN SRIDHAR
Date:	11-12-2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Securing 5G IoT Networks: A Machine Learning Framework for Zero-Trust Intrusion Detection System

Preetham Charan Sridhar
x23183683

Abstract

The integration of 5G and IoT devices has completely changed industries, where this seamless integration made the devices perform faster data transmission, real-time automation, and seamless device connectivity. However, this revolution also introduced major security challenges in the real world. The research addresses the challenges faced by the traditional setup by building advanced Intrusion Detection Systems (IDS) using machine learning techniques. Hybrid models like DT-CART which is combined with XGBoost have shown high performance with an accuracy of 99%. The research further provides an in-depth analysis and key findings of securing the 5G IoT networks by combining machine learning, hybrid models, and federated learning. The Federated Ensemble with Stacking and Majority Voting has achieved an accuracy of 91%, proving that the system can identify and respond to malicious activities in a distributed environment rather than keeping them stored or processed in a central server. The Federated learning was further explored by integrating Zero-Trust principles. Dynamic trust scores were used to exclude the untrusted clients, where this method makes sure that only the trusted clients will contribute to the global model results in reducing security risks like adversarial predictions and data poisoning.

1.1 Introduction

The growth of the 5G technology and the expansion of IoT devices in everyday life have changed industries and increased efficiency. 5G technology provides fast communications, low delay rates, and smooth connection between IoT devices (Ahmad *et al.*, 2018). These IoT networks are distributed and large-scale in nature so in which introduces new critical challenges. This study focuses on creating a better Intrusion detection system to solve security issues in the 5G IoT networks to make networks safe and reliable. (Ahmid and Kazar, 2023)

1.2 Background and Importance

With the arrival of 5G, this growth of using IoT devices in different fields began to grow very fast due to their better communication speed and latency at very low levels. The fields of healthcare, transportation, and manufacturing show a high dependency on IoT systems for real-time data interchanges and automation. However, this major spread and IoT's large-scale nature have made these industries get targeted more by cyber criminals. Due to the insufficient way of securing the IoT networks using the traditional setup, hackers take advantage of the weak points of IoT systems, like poor encryption, outdated software, and weak access controls, to perform attacks like DDoS, ransomware, and data breaches. Still, most industries are using

traditional security tools like firewalls and antivirus programs which are not enough to tackle the changing and advanced threats targeting IoT networks. (Virat *et al.*, 2018) The intrusion Detection system which is powered by machine learning and hybrid models, offers a better solution, where they help to monitor the network traffic across the devices to detect any unusual patterns, and reduce risk in real-time overall. This system should be used for the critical 5G networks, due to the more complex and many connected devices, and these systems can work better in handling the threats. Securing the network is the most important thing which protects the sensitive data and keeps operations running smoothly. (Pinto *et al.*, 2023)

1.3 Motivation

The need for 5G has become more crucial for the applications which are running critically, as it is required to have a strong security systems. Traditional methods like authentication and encryption mechanisms with various access controls for securing the IoT devices are not sufficient enough, where in the absence of a zero-trust approach in 5G IoT deployments. The main aim of this work is to develop a sophisticated Intrusion Detection System (IDS) for 5G Internet of Things (IoT) networks, characterized by interconnection of billions of devices that communicate sensitive information.The research focuses on developing an advanced Intrusion Detection System that can detect and classify malicious activities in IoT networks, by using advanced machine algorithms, which include hybrid approaches, and these are done using techniques like Decision Trees (DT), Random Forest (RF), and boosting algorithms like XGBoost.(Mittal and Batra, 2022) The research also aims to create an IDS that can also address the real-time threats in resource-constrained environments. The study also evaluated the use of federated learning which provides security in distributed edge-based 5G IoT systems, and also further explored the concept of integrating Zero Trust principles with Federated Learning which works by only the trusted clients are allowed to contribute to the global model, which reduces the risks like adversarial attacks and data poisoning in distributed 5G IoT networks.(Zhang *et al.*, 2021) This combination of Federated Learning and Zero Trust adds a huge advantage to the overall security framework by verifying the process of the client's trustworthiness and ensuring robust protection for sensitive data in dynamic IoT environments.

1.4 Research Question

1. How accurately hybrid models and basic classifiers can detect multi-classification attack types?
2. How federated learning integrated with Zero Trust principles helps in improve security and scalability in IoT networks?

1.5 Summary of Contents

The report section includes the **Related research**, where past studies on Federated Learning and Zero Trust Intrusion Detection conducted by various researchers are reviewed, and areas for further research are identified.

The **Methodology** section really gives an overall view of the strategies to be followed in order to achieve the objectives of the study, with detailed processes explained under the **Design Specification.** The following sections of **Implementation**, **Evaluation**, and **Discussion** give the detailed analysis of tools, and frameworks which is followed and the experimental results. Finally, the report concludes with reflections from the results and highlights future avenues of work.

2 Related Work

2.1 Literature Review

(Lanka, Aung Win and Eshan, 2021) showcases that combining edge computing with 5G technology improves the data processing on the IOT devices which benefits in reducing the latency and improves overall network performance. Utilizing Egde computing helps in processing the data much closer to where the data was generated, which leads to speeding up the response time. However, it also has some bottlenecks that involve limitations in both storage and security risks present within multi-device connectivity. While this work provides strong evidence on which to find and investigate the benefits involved, edge computing needs deeper research in order to perform the enhancement of storage capacities together with security for diverse connected devices.

(Ahmad *et al.*, 2018) focused on the key security challenges involved in the deployment of the 5G networks with the potential solutions to overcome them and also showcased the security risks involved in network slicing, privacy concerns, and denial-of-service (DoS) attacks. However, the research paper failed to address the real-world testing for the proposed security measures, and the provided solutions are not application-specific. It is suggested that the paper looks more general which makes it hard to tackle the evolving threats in 5G networks.

(Seraphim *et al.*, 2018) have explored several machine learning methods for detecting intrusions in networks and finally, they found that techniques like Mini Batch K-Means and autoencoders (which is one of the deep learning tools) have achieved a high accuracy rate, where one model reached 97.85. Although the paper had gaps in exploring advanced models for detecting the new and complex attacks for making the systems detect faster in real-time situations. It has been reported that this improvement like real-world testing and using advanced models for detecting helps in working better with modern network challenges.

(Virat *et al.*, 2018) have proposed a 5-layered model which is a layered architecture of IoT for addressing the security and privacy challenges which are unique to IoT. The study concerned more about the issues related to data privacy, integrity, and specifically highlighting the DDOS attack-related risks, eavesdropping, and malicious code attacks. However, the study have left the gap in providing the specific mitigation solution for which they have highlighted and had only limited focus on providing the implementation of real-time detection or response mechanisms. The evidence suggests that, addressing these downsides and delving into technical implementation can help developers who are working on real-world IoT security.

(Ramezanpour and Jagannath, 2022) has introduced an intelligent zero trust architecture (i-ZTA) for 5G/6G networks for addressing the security issues in an untrusted environments. They showcased the AI-driven framework which uses the real-time monitoring, dynamic policy decisions, and risk assessment for secure access control. The challenging part of the i-ZTA is the complexity is high and the processing cost for large data in real-time is huge when using smaller devices. It appears that the i-ZTA is a solid framework, but addressing the scalability and resource demands is very crucial while working on real-world deployments.

(Cao *et al.*, 2020) have done a detailed survey on the security aspects of 3GPP 5G networks, which they deeply focused on features like IoT integration, Device-to-Device (D2D) communication, and network slicing. They have also addressed the vulnerabilities related to D2D privacy and IoT data protection and also covered some of the existing solutions and research areas. However, the paper lacks in real-time security mechanisms and mitigation strategies within the 5G environment and there are no proper details on the part of the practical implementation. This analysis indicates that the report was good in addressing the 5G security issues, but practically addressing them by adding real-world examples would have been more helpful.

(Al-Juboori *et al.*, 2023) used machine learning algorithms like XGBoost, Random Forest, Gradient Boosting, and Decision Tree for detecting the Man-in-the-Middle (MTM) and Denial of Service (DoS) attacks on IoT networks and was able to achieve an accuracy of above 97% for both the attacks. The major limitation was only focusing on the static machine learning models for detecting the attacks, and as we know IoT environments are dynamic, lack of adaptability to new evolving attack patterns was majorly missing. It is believed that the paper needs to address the need for real-time response and the chosen models work only on the provided dataset and will struggle on complex threats in live environments.

(Cui *et al.*, 2018) have discussed how machine learning can be useful in analyzing network traffic, identifying devices, and improving security. They worked on different models to show how the ML tools helped to make IoT systems smarter and more efficient. However, the downside of the paper is that it does not say how well ML works in scalability situations when there is an increased number of connected devices. There is a lack of detail regarding privacy concerns during machine learning analytics in IoT. The perspective is that, the paper has a solid overview of machine learning applications for IoT, but addressing the major concerns like data privacy, scalability, and energy efficiency concerns will have been better.

(Abdalzaher *et al.*, 2023) reviewed the machine learning role in securing IoT-based smart systems, which are used in the smart campuses and earthquake warning systems. They have explored various machine learning models for improving IoT security, which includes linear and nonlinear models. However, the paper doesn't address the real-time adaptability and scalability during high-traffic IoT environments, which is very important during responsive security. It is appears that considering these details will have been more helpful during the practical IoT Security.

(Alsaedi *et al.*, 2020) Created a TON_IoT telemetry dataset which was specifically created for training and evaluating Intrusion Detection Systems (IDS) in IoT and Industrial IoT (IIoT) applications. The dataset included all the operating system logs, telemetry logs, and network traffic data which was represented exactly like a realistic testbed for a variety of cyber-attack scenarios. However, the key limitation was in real-world testing scenarios. It appears that this is the synthetic dataset, we need more testing in the real conditions would be helpful in IoT security research.

(Dorogush, Ershov and Gulin, 2018) The study compared three popular Gradient Boosting Decision Tree (GBDT) algorithms XGBoost, LightGBM, and CatBoost, focusing on the CPU performance in terms of accuracy, speed, reliability. On the results, the LightBGM performed well overall in terms accuracy and speed, which is followed by XGBoost and CatBoost was the poor performer compared all three. The limitations in the study is the author used mid-range hardware for the testing, I believe there is no full performance is reflected there shoud be all three range hardwares (Low, Mid, High) should have been included in the testing so it would have provided more comprehensive evaluation of these algorithms.

(Mamoun Alazab et al., 2011) proposed a solution for detecting zero-day malware using supervised machine learning algorithms focused on API call signatures. They evaluated with various algorithms like Naive Bayes, k-Nearest Neighbor, and Decision Trees, and achieved a high accuracy rate surpassing 98.5% in true positive detection. The limitation of the solution is that, the system was not able to adapt new methods that hackers used for hiding the malware which missed past their API-based detection approach. According to this research, the learning methods should improve on these tricks to make the system work better at catching advanced, changing malware.

(Javeed *et al.*, 2024) The study on the Federated Learning (FL) based Zero Trust IDS combining CNN and BiLSTM models for IoT networks has been focused mainly on privacy, accuracy, and scalability. The model has achieved a high accuracy rate of 99.99% on both the datasets CICIDS2017 and Edge-IIoTset. However, the limitation of the study is, that the authors have focused more on the ICT datasets. The perspective is that, using more diverse OT datasets could have improved applicability for industrial IoT security, which should have provided better evaluation.

(Asad and Otoum, 2024) The research proposed an integration of Federated Learning (FL) and Zero-Trust security for wireless networks, which addresses the privacy and security challenges faced by modern cyber attacks. This approach uses federated learning to process the data across multiple locations or devices rather than in a single, central place. It also has been combined with zero trust for controlling access, which tends to build a strong defense system. However, the limitation in practical implementation should have been more valuable in the real-world validation. It is suggested that, authors could have used any OT datasets to show some useful insights, it could have been more effective.

(Nour, M. 2023)The research highlights the growing risk in 5G IOT networks in resource-constrained environments. The study proposed a Smart Zero-Trust Framework, which is very efficient and effective in detecting threats quickly in real time by combining machine learning with Zero Trust principles. However, the key limitation is that the research provided more insights into the binary classification of malicious traffic across various IoT industries, so exploring multi-class classification attacks would be a fruitful direction and additionally exploring hybrid models can lead to improved accuracy and speed in IoT intrusion detection systems. The perspective is that the research could have explored multiclassification instead of working binary attacks on all four datasets.

S. No	Authors	IDS System for Operational Technology	Review Comments	Accuracy and Other Evaluation Parameters
1	Mohamed G. Nour (2023)	Smart Zero-Trust Framework for 5G IoT Networks	Focused on Binary attack but lacks in multi-class insights	Achieved high effiencity on the binary attacks but lacks in multi-class performance.
2	Lanka, Aung Win, and Eshan (2021)	Edge Computing Integration with 5G for IoT Devices.	Shows the latency reduction but missed focusing on the storage and security issues.	No accuracy reported, emphasizes reduced latency and improved data proximity.
3	Ramezanpour and Jagannath (2022)	Intelligent Zero Trust Architecture (i-ZTA) for 5G/6G Networks	Higly complex on the real time data on smaller devices.	Focuses on scalability and resource demand concerns.
4	Javeed et al. (2024)	Federated Learning-based Zero Trust IDS combining CNN and BiLSTM for IoT Networks	Limited to ICT datasets, no exploration to industrial IoT.	Achieved 99.99% accuracy on CICIDS2017 and Edge-IIoTset. Failed to address OT dataset evaluation.

Table 1. Sumary of research literature review.

3 Research Methodology

3.1 Dataset Source and Collection

CICIOT2023 (https://www.unb.ca/cic/datasets/iotdataset-2023.html) – The dataset is from the CIC (Canadian Institute for Cybersecurity) dataset series which is focused on IoT-specific network traffic, which contains over 46 network traffic features, with multiclass labels for finding the different types of attacks. The classes include DoS, DDoS, Reconnaissance, Brute

Force, Mirai, and Spoofing alongside benign traffic. This dataset has been used to evaluate the classifiers performance in detecting the diverse attack scenarios in IoT environments.

BoT-IoT (https://research.unsw.edu.au/projects/bot-iot-dataset) - The dataset is developed by the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS), the dataset majorly focuses on the broad coverage of the IoT network traffic and a wide range of attack scenarios. The dataset includes variety of the feature sets. The Attack scenarios in the dataset includes DDoS, data exfiltration, keylogging, and service scanning, which reflects diverse threats faced by 5G IoT networks. Bot_Iot is broadly used dataset for its coverage of attacks in context with the 5G IoT networks.

3.2 Dataset Preparation and Initial Processing

The **CICIoT2023** dataset was uploaded to the Google Drive and loaded into the Google Colab Pro environment. At first, columns were renamed to maintain consistency and missing was checked for the data quality. As since there were no missing values found, unnecessary columns like protocol_type were dropped to get better performance during the model training. To address the class imbalance in the dataset, SMOTE technique was applied for minority classes to achieve a balanced dataset. Standardscaler has been used to feature scaling in the dataset to bring all numerical values to a similar range. Then the dataset was split into training and testing sets with an 80:20 ratio using the train_test_split function. Stratification was used to maintain the class distribution consistent in both the training and testing sets

The **BoT_IoT** dataset was prepared for the study, consisting of training dataset (293,437 rows), and the testing dataset (73,735 rows). Initially, the dataset was analysed to find the irrelevant columns, such as protocol details, source/destination IPs, and unnecessary identifiers, was removed to focus only on the important features. The Class distribution analysis showed that the dataset was highly imbalanced, in particular for rare attack categories like Keylogging and Data_Exfiltration. To address the issue of class imbalance, undersampling and oversampling techniques were used to maintain the majority and minority classes distribution. Subcategories like TCP and UDP were merged into a broader category named DoS&DDoS, to simplify the classification task. After balancing the datasets were split into training (80%) and testing (20%) sets for consistent evaluation.

3.3 Feature Engineering and Selection

CICIoT2023 - The Feature Engineering and Selection was performed to optimize the dataset for the classification task. To improve the consistency, columns were reviewed and cleaned. Heatmap was generated to remove the redundant features correlation. Attack categories were grouped into broader labels and converted into numeric format using label encoder.

Bot_IoT – Feature Engineering and Selection was done to improve the performance of the machine learning models. Initially, missing values were checked and irrelevant columns removed to focus on the important features. The BoT_IoT dataset was highly imbalanced in attack types, undersampling techniques were used for the majority classes and SMOTE was used for the minority classes to make the dataset as balanced. Standard scalers were used to scale the numerical features to maintain the same range. The sub_category column was chosen

as the target variable to represent different attack types. Finally, the correlation heatmap was generated to ensure the best data has kept for the training.

3.4 Model Selection and Training

Base Models - The Base Models includes Logistic Regression, K-Nearest Neighbors, Random Forest, Naive Bayes, Decision Tree, and Support Vector Machine helped as the foundation for benchmarking the performance of advanced techniques.

Advance Hybrid Models - To Improve the performance, hybrid models has been used by combining the decision trees with boosting techniques.

- **Tuned Hybrid Model –** The model helps in achieving the better accuracy and can handle patterns which are complex with the combination of a simple decision tree and the boosting power of XGBoost.
- **Hybrid Stacking Model –** The model uses a stacking method, where there will be combined prediction from the Decision Trees and XGBoost and it is refined by the logistic regression to improve the overall results. (Sajid *et al.*, 2024)
- **Stacking Ensemble –** The method combines multiple models, to make the IDS more consistant and accurate for the multi-class classification tasks.

Federated Learning Approaches - Federated Learning was implemented for addressing the issues and challenges in the decentralized environments and privacy concerns in data sharing.

- **Basic Federated Learning –** The method helps in testing, how well the model is trained on the different dataset without combining them into one central dataset.
- **Federated Ensemble with SMOTE –** The method helps in addressing the uneven class distribution in the data to balance dataset by creating a synthetic samples, which makes sure that model trainig is better during the federated environments.
- **Federated Ensemble with Stacking and Majority Voting –** The stacking and majority voting are the two powerful techniques to improve the IDS in the distributed systems.
    - **Stacking –** During the stacking technique, predictions from multiple models (e.g., Decision Trees, Random Forest, and XGBoost) are combined and these predictions will be passed on to the meta-learner which is the logistic regression, and it refines the final output. This makes the model better across the diverse data scenarios.
    - **Majority Voting –** The majority voting approach involves in combing the predictions from the multiple models and selects the most predicted class as its final decision. (Nguyen and Beuran, 2024)

3.5 Zero Trust Integration in Federated Learning

This section explains how Zero Trust principles were integrated into the Federated Learning framework to increase security and privacy. Federated Learning trains multiple clients collaboratively on a machine-learning model without sending raw data. However, this distributed nature will make the Federated Learning model vulnerable to malicious clients or

corrupted data. By Implementing a zero-trust framework, only trustworthy client contributors are accepted to send the data to the central server adhering to the principle of "never trust, always verify". (Liu et al., 2024)

- **Client Identity Verification –** Initially, a unique identifier (Client ID) will be assigned for each client for the account and access control. This ensures that only the authorised client can be participated during the training process, where this makes sure that there is not unauthorized or rogue devices are contributing to the system.
- **Data Encryption –** All the communication between the server and client are encrypted to protect the data during the transmission. Where this helps in avoiding unauthorized access or tampering during transit.
- **Dynamic Trust Scores Mechanism –** The Trust score mechanisms is performed to achieve the reliability during the training from each client. Initially, all client are given high scores, and these scored are graudually updated based on their accuracy of predictions and behaviour (e.g., data integrity during decryption). The client who posses low trust scored will be flagged as untrustworthy and eliminated from the aggregation process. (ZhangYifei et al., 2024)
- **Simulated Intrusions –** To evaluate the performance of the framework, various attack simulations were performed,
    - **Data Posioning -** For testing the method, corrupted data was deliberately assigned to the client, resulting in an accuracy drop, which was reflected in a reduced trust score.
    - **Failed Decryption –** During the testing, client's data integrity was set to 0 which resulted in the decryption fail.
    - **Adversarial Predictions –** During the testing, a client was programmed to send random predictions for the central test set, which resulted in accuracy drop, leading to exclusion. (Wu et al., 2023)
- **Trust-Based Aggregation –** During the aggregation, the client's who have the trust score above the threshold can able to make the final prediction, this makes sure that the final model is not influenced by malicious or untrustworthy clients.

3.6 Model Evaluation

In the research, the performance of the base models, Advance Hybrid Models and Federated Learning Approaches have been calculated using various metrics, cross-validation techniques, and confusion matrix analysis. In federated learning setup, to ensure the exclusion of the untrusted clients, the trust scores were recalculated dynamically. Below are the main objectives of the model evaluation process,

Performance Metrics - The model were evaluated using the Accuracy, Precision, Recall, F1-Score, and Prediction time (seconds) to measure the model's ability to correctly classify the multi-class IoT attack traffic. These metrics shows that how well the models have handled the balanced and imbalanced datasets.

- Accuracy = (TP + TN) / (TP + TN + FP + FN)
- Precision = TP / (TP + FP)

- Recall = TP / (TP + FN)
- F1 Score = 2 × (Precision × Recall) / (Precision + Recall)

Cross Validation - The K-fold cross-validation method was used as part of the evaluation method to check how well the model performed. During the method, the dataset is divided into smaller folds which can be 3 or 5, and it will be trained on some folds and tested on the remaining folds, where this method ensures a more detailed and reliable evaluation, and also this methods helps in reducing the chances of the overfitting and make sure that the results are dependable.

Trust-Based Filtering Evaluation - To evaluate the performance of the framework, various attack simulations were performed like Data Posioning, Failed Decryption, and Adversarial Predictions and the trust score's predefined threshold (e.g., 0.7) below that can lead to client exclusion.

$$T = w_1 \times \text{Accuracy} + w_2 \times \text{Data Integrity} + w_3 \times \text{Consistency}$$

Fig 1. Trust Score Formula

On comparative analysis results before and after trust filtering demonstrated,

- **Without Trust Filtering** – On an untrust filtering the accuracy was at 91% which is susceptible to poisoned and adversarial inputs.
- **With Trust Filtering** – On an trust filtering the accuracy was increased to 98% which proved the effectiveness of trust-based client validation. (Tariq et al., 2024)

4 Design Specification

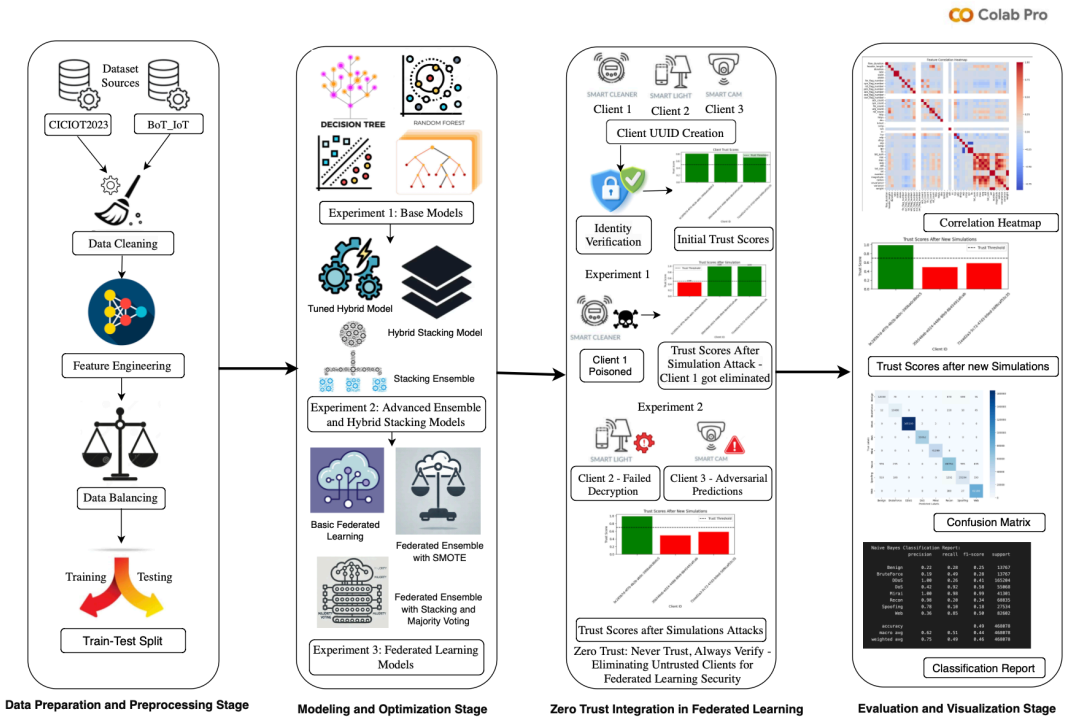

Fig 2. Design Flow

- **Data Preparation and Preprocessing –** This step will include data cleaning, feature engineering, data balancing, and the train-test split. The raw data repository is in CSV format and stored in Collab Pro.
- **Modeling and Optimization –** It comprises three experimental settings: base models, Advanced Ensemble/hybrid Stacking Models, and federated Learning models. Such approaches and architectures are realized using CICIoT2023 and Bot_IoT datasets.
- **Federated Learning with Zero Trust Integration** – Zero Trust principles are integrated into the federated learning process by dynamically calculating trust scores for each client. The clients are excluded when they fall below the defined threshold limit which prevent the global model from the malicious or rogue clients contribution.
- **Evaluation and Visualization** – For the model evaluation, metrics are used like accuracy, F1 score, Recall, precision, and confusion matrix, and the results are represented in graphs and tables for better understanding.

5 Implementation

5.1 Software and Hardware Used.

The Integrated Development Environment (IDE) used for this project is Google Colaboratory (Colab), specifically the Pro version, which has high-RAM environments and faster GPUs/TPUs to run the resource-intensive machine learning tasks. The research configuration manual of the project has detailed information on the hardware and software.

5.2 Libraries Imported and used

Pandas, Numpy, Sklearn, imblearn, Seaborn, Matplotlib.

5.3 Data Preparation and Preprocessing

- **Dataset Loading and Initialization** – The datasets CICIoT2023 and Bot_IoT were uploaded to the Google Colab Pro for preprocessing process. For consistency, the columns have been renamed for consistency.
- **Data Cleaning –** The columns which are irrelevant were identified and removed to focus only on the relevant features.
- **Handling Class Imbalance –** For CICIOT2023 dataset, the SMOTE was applied to oversample minority classes, and for BoT_IoT udersampling and SMOTE were used to balance the dataset.
- **Feature Engineering** – Attack categories were grouped into broader labels for classification and used StandardScaler to standardize the Numerical features to have an consistent range for all the features
- **Train-Test Splitting and Validation -** Both datasets were split into 80% training and 20% testing sets and the heatmap has been generated to check the feature correlation.

5.4 Model Training Process

- **Base Models –** All base models were evaluated using 5-fold cross-validation to ensure effectivness and generalizability.
- **Advance Hybrid Models** – **(Tuned Hybrid Model)** A hybrid approach, at first, the DecisionTreeClassifier (CART) was trained and the predictions were used as additional features for XGBClassifier. To optimize the performance of both the components Hyperparameter tuning was used. **(Hybrid Stacking Model)** CART and XGBoost were trained as base learners. Then the prediction were passed to the LogisticRegression meta-learner which combined them to make final predictions. **(Stacking Ensemble)** Mutiple base modes were trained (XGBoost, Random Forest, Decision Tree), Then the prediction were passed to the LogisticRegression meta-learner which combined them to make final predictions.
- **Federated Learning Approaches** – **(Basic Federated Ensemble)** The dataset has been divided into three clients and each clients were receiving a portion of the training data, and the each client was trained its own DecisionTreeClassifier, GradientBoostingClassifier, and AdaBoostClassifier. StackingClassifier was used to combine the each model with the LogisticRegression meta-learner. The prediction from the each client were aggregated using majority voting. Performance metrics were used to compute the assessment. **(Federated Ensemble with SMOTE)** To balance class distribution of the dataset, SMOTE was applied before model training, and as a fallback Random Oversampling was used. Each client was trained its own DecisionTreeClassifier, GradientBoostingClassifier, and AdaBoostClassifier. The prediction from the each client were aggregated using majority voting. Performance metrics were used to compute the assessment. **(Federated Ensemble with Stacking and Majority Voting)** The dataset has been divided into three clients and each client were trainined using a StackingClassifier consisting of DecisionTreeClassifier, GradientBoostingClassifier, and AdaBoostClassifier as base models. Majority voting was applied to the predictions from the stacking models of all clients to determine the final labels. Performance metrics were used to compute the assessment.

5.5 Zero Trust Integration in Federated Learning

This section shows the dynamic trust score mechanism integrated into the federated learning system to implement Zero-Trust principles. The Trust score calculated for each is based on the formula. If any client has a trust score below the threshold limit, the client will be excluded as an untrusted client and can not contribute to the global model. During the implementation, two experiments were conducted to check how Federated learning continuously validates client contributions and excludes potentially harmful participants

Results and Observations - The results showcase that the model is highly effective and that only the Trusted clients are contributing to the federated learning, this ensures that the system retains its reliability during the dynamic recalibration of trust scores by excluding the compromised clients.

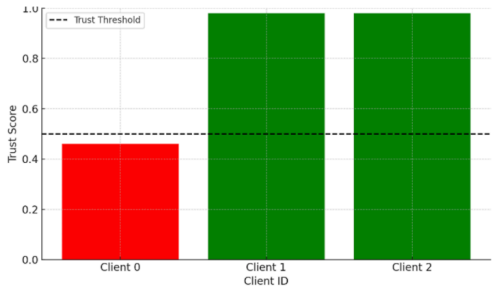

Fig 3. Experiment 1 - Trust Scores after Client 0 Poisoned

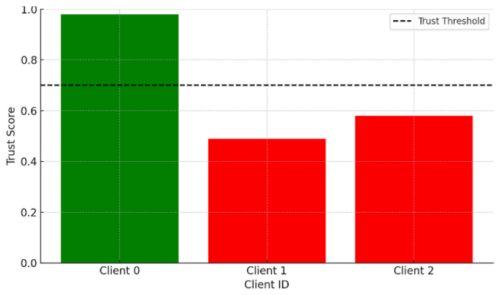

Fig 4. Experiment 2 - Trust Scores after Client 1 (Failed Decryption) and Client 2 (Adversarial Predictions)

6 Evaluation

6.1 Case Study - CICIOT2023

Experiment 1 - Base Models - As of base classifiers results, K-Nearest Neighbors (KNN) has achieved a higher accuracy of 0.99, which showed best among all the classifiers to classify the data accurately, however, the prediction time was significantly high with an average of 1214.87 seconds which indicated a trade-off between accuracy and efficiency. The random forest has been a balanced choice between performance and speed with the metrics 0.93. KNN has been best for its time-sensitive choice, as it has the lowest prediction time. Among all the classifiers, Naive Bayes will be the weakest performer, which shows the limitation in the complexity of the dataset, and the last decision tree will be the reliable and efficient classifier for this dataset. The Cross-validation results of the base classifiers show that the performance was consistent and reliable. KNN has the best accuracy of 0.98 across all the classifiers, which is followed by the random forest and decision tree. Naive Bayes has performed worst among all with an accuracy of 0.49 which makes very less suitable for this task. The results show that KNN and Random Forest are very effective for the classification.

Base Classifiers					
Evaluation/CV Paramters	**Logistic Regression**	**K-Nearest Neighbors**	**Random Forest**	**Naive Bayes**	**Decision Tree**
Accuracy	0.77	0.99	0.93	0.49	0.82
Precision	0.78	0.99	0.94	0.75	0.84
Recall	0.78	0.99	0.93	0.49	0.82
F1 Score	0.77	0.99	0.93	0.46	0.82
Prediction Time (S)	0.0599	1214.87	0.3683	0.4651	0.0854
Cross – Validation Scores					
CV Scores	[0.7732, 0.7680, 0.7694]	[0.9868, 0.9868, 0.9866]	[0.9290, 0.9289, 0.9317]	[0.4919, 0.4965, 0.4962]	[0.8193, 0.8192, 0.8193]
Average CV Accuracy	0.77	0.98	0.92	0.49	0.81

Table 2. Evaluation Paramters and Cross-Validation Score Comparision.

Experiment 2 - Advanced Ensemble and Hybrid Stacking Models - The evaluation of advanced ensemble and hybrid stacking models shows a balanced performance between accuracy and efficiency. The Tuned Hybrid Model is best suitable for real-time application due to its balanced performance between the accuracy of 0.98 and the prediction time of 1.45 seconds. The Hybrid Stacking Model has achieved the higher accuracy performance among all and also excels in prediction time of 0.1931 seconds, which makes it appropriate for real-time or batch processing tasks. At last, the Stacking Ensemble performed exactly similarly to the Hybrid Stacking Model also shows the average prediction comparatively. These results show the importance of which models should be selected based on the specific requirements of accuracy and prediction speed for a given application. The Cross-Validation Scores results Hybrid Stacking Model achieves the highest mean accuracy of 0.9801, showcasing the consistency across the multiple folds. On opposite, the tuned Hybrid model is slightly behind with the mean value of 0.9737, shows the lower consistency compared to the hybrid stacking model. On the whole, stacking ensemble have shown low mean accuracy at 0.9202, which indicating a trade-off between its simpler architecture and predictive power. The results shows, the Hybrid Stacking Model would be the preferred choice for scenarios prioritizing accuracy and consistency

Advanced Ensemble and Hybrid Stacking Models			
Evaluation Parameter	**Tuned Hybrid Model (DT-CART + XGBoost)**	**Hybrid Stacking Model (DT-CART + XGBoost + LR meta-learner)**	**Stacking Ensemble (XGBoost, RF, DT, LR meta-learner)**
Precision	0.98	0.98	0.98
F1-Score	0.98	0.98	0.98
Recall	0.98	0.99	0.99
Accuracy	0.98	0.98	0.98
Prediction Time (S)	1.45	0.1931	0.4797
Cross – Validation Scores			
CV Scores	(0.9753, 0.9730, 0.9728)	[0.9801, 0.9800, 0.9802]	[0.9228, 0.9133, 0.9245]
Mean CV Accuracy	0.9737	0.9801	0.9202

Table 3. Performance and Cross-Validation Score Comparison of Advanced Ensemble and Hybrid Stacking Models.

Experiment 3 – Federated Learning - The results of FL with stacking and majority voting have achieved a higher accuracy of 90% with equal strong performance on all the metrics. This method has proved to be the most reliable for handling distributed data and reducing errors. The other two methods have failed to match the effectiveness of the advanced ensemble techniques, which may be due to the overfitting or noise from the synthetic data generation.

Overall, the results show that Federated Ensemble techniques have improved the effectiveness and security.

Evaluation Parameter	Basic Federated Leraning	Federated Ensemble with SMOTE	Federated Ensemble with Stacking and Majority Voting
Accuracy	0.88	0.83	0.90
Precision	0.89	0.87	0.91
Recall	0.88	0.83	0.90
F1 Score	0.88	0.84	0.90

Table 4. Federated Learning Comparison Table.

6.2 Case Study 2 – Bot_IoT

Experiment 1 - Base Models - The experiment was done using the Base classifiers and were evaluated using various metrics, which includes accuracy, precision, recall, F1 score, and average prediction time. KNN and Random Forest have achieved a high accuracy rate of (0.98) with KNN showcasing the results with the lowest prediction time 0.00115 seconds. Naive Bayes can used for speed-prioritized applications as it has the fastest prediction time overall. SVM shows its superior performance of achieving 0.89, but there was an average length prediction time which indicates a trade-off between accuracy and efficiency. Cross-validation is done to check the stability of each classifier across the multiple folds. The Decision Tree has achieved the highest average cross-validation score (0.988), which is followed by the KNN (0.98) and Random Forest (0.983) indicating strong consistency and reliability. The results below show that the Decision Tree, KNN, and Random Forest are the top performer classifiers which indicated the balance between the accuracy and stable performance across folds.

Base Classifiers						
Evaluation/CV Paramters	**Logistic Regression**	**KNN**	**Naive Bayes**	**Random Forest**	**SVM**	**Decision Tree**
Accuracy	0.86	0.98	0.71	0.98	0.89	0.94
Precision	0.79	0.98	0.76	0.98	0.88	0.94
Recall	0.86	0.98	0.71	0.98	0.89	0.94
F1 Score	0.81	0.98	0.70	0.98	0.87	0.94
Prediction Time (S)	0.00569	0.00115	0.00018	0.02	38.78	0.02
Cross – Validation Scores						

Cross-Validation Scores	[0.8659, 0.8656, 0.8673, 0.8677, 0.8657]	[0.9829, 0.9835, 0.9824, 0.9822, 0.9841]	0.7039, 0.6947, 0.7068, 0.7050, 0.7197]	[0.9825, 0.9848, 0.9841, 0.9829, 0.9848]	[0.8676, 0.8668, 0.8687, 0.8698, 0.8661]	[0.9880, 0.9887, 0.9881, 0.9885, 0.9881]
Average CV Score	0.86	0.98	0.70	0.983	0.867	0.988

Table 5. Evaluation Paramters and Cross-Validation Score Comparision.

Experiment 2 - Advanced Ensemble and Hybrid Stacking Models - The results of the advanced ensemble and hybrid stacking models has enhanced the intrusion detection performance. The Tuned Hybrid, Hybrid Stacking, and Stacking Ensemble models achieved an highest accuracy, precision, recall, and F1-score of 0.98. In this experiment, the Tuned Hybrid Model is the fastest at 0.0551 seconds. The Cross-validation is done to achieve model stability and broad applicability, cross-validation was implemented. The tuned hybrid model achieved the highest mean cross-validation accuracy scores, which shows balances in performance and efficiency effectively. This experiment shows the detection capabilities where the Tuned Hybrid Model was well suited for its time-sensitive intrusion detection applications.

Advanced Ensemble and Hybrid Stacking Models			
Evaluation Parameter	**Tuned Hybrid Model (DT-CART + XGBoost)**	**Hybrid Stacking Model (DT-CART + XGBoost + LR meta-learner)**	**Stacking Ensemble (XGBoost, RF, DT, LR meta-learner)**
Precision	0.98	0.98	0.98
F1-Score	0.98	0.98	0.98
Recall	0.98	0.98	0.98
Accuracy	0.98	0.98	0.98
Prediction Time (S)	0.0551	0.10	0.22
Cross – Validation Scores			
Cross-Validation Scores	[0.9928, 0.9945, 0.9942, 0.9940, 0.9935]	[0.9886, 0.9899, 0.9901, 0.9895, 0.9896]	[0.9878, 0.9889, 0.9891, 0.9884, 0.9886]
Mean CV Accuracy	0.9938	0.9895	0.9886

Table 6. Performance Comparison and Cross-Validation Score Comparsion of Advanced Ensemble and Hybrid Stacking Models

Experiment 3 – Federated Learning - In Federated learning experiment, Basic Federated Learning has achieved an overall 0.90, which shows a balanced performance across metrics. Federated Ensemble with SMOTE shows an accuracy of 0.83 but there is a slight decrease in the recall and f1 scores, which shows that SMOTE did help with precision but at the cost of

overall accuracy.The most advanced model Federated Ensemble with Stacking and Majority Voting shows the highest accuracy (0.91) among all the models which shows the most reliable performance across all metrics.

Evaluation Parameter	Basic Federated Learning	Federated Ensemble with SMOTE	Federated Ensemble with Stacking and Majority Voting
Accuracy	0.90	0.83	0.91
Precision	0.90	0.91	0.90
Recall	0.91	0.84	0.91
F1 Score	0.90	0.85	0.90

Table 7. Federated Learning Comparison Table.

6.3 Case Study 3 – Zero Trust Integration in Federated Learning.

The below table showcases the effectiveness of the Zero Trust framework in addressing Client 0 poisoning within federated learning. This led client 0 to get excluded from the global model contribution to maintain the system reliability.

Client ID	Accuracy	Data Integrity	Consistency	Threshold	Trust Score	Status
Client 0	0.1	1.0	1.0	0.5	0.46	Untrusted
Client 1	0.98	1.0	1.0	0.5	0.98	Trusted
Client 2	0.98	1.0	1.0	0.5	0.98	Trusted

Table 8. Experiment 1 - Client 0 Poisoned

The table shows the efficacy of the Zero Trust integration in handling scenarios involving failed decryption (Client 1) and adversarial predictions (Client 2). Due to the simulation attacks on client1 and client2, the dynamic trust score mechanism recalibrates scores based on accuracy, and data integrity and made the clients with low trust scores below the defined threshold has been excluded. This made sure that only trusted clients could contribute to the global model.

Client ID	Accuracy	Data Integrity	Threshold	Trust Score	Status
Client 0	0.98	1.0	0.7	0.99	Trusted
Client 1	0.98	0.0	0.7	0.49	Untrusted
Client 2	0.166	0.98	0.7	0.58	Untrusted

Table 9. Experiment 2 - Client 1 (Failed Decryption) and Client 2 (Adversarial Predictions)

The table shows the impact of trust filtering by comparing performance metrics with and without the filtering mechanism. With Trust filtering the system accuracy has showcased accuracy 98% proving how reliable and effective the system is. The dynamic recalibrating of

trust scores has improved the performance across all the metrics by filtering only the trusted clients to contribute to the global model.

Metric	Without Trust Filtering	With Trust Filtering
Accuracy	91%	98%
Precision	90%	98%
Recall	91%	98%
F1-Score	90%	98%

Table 10. Performance Comparison With and Without Trust Filtering.

6.4 Discussion

Experiment (Base Models) - The base models have demonstrated the diversity of performance among the models on both datasets where the highest accuracy of 99% on the CICIOT2023 dataset and on the Bot_IoT dataset by KNN of 98%. Nonetheless, the prediction time was much higher, making it less appropriate for time-critical applications. Random forest is the balanced choice between good accuracy and moderate prediction time across both datasets. These results show the need for hybrid models or ensemble models to address the compromise between accuracy and efficiency. **Experiment (Advanced Ensemble and Hybrid Models) -** On both the datasets, the Hybrid Stacking Model has achieved the highest accuracy of 99%, however, the prediction time was very high making it suitable for offline works. The Tuned Hybrid Model has a balanced accuracy (98%) and a prediction time was (1.45) seconds which makes it more suitable for real-time applications. The stacking Ensemble model shows lagged accuracy compared to both models. **Experiment (Federated Learning Approaches) -** On both the datasets, the Federated Ensemble with Stacking and Majority Voting showed significant performance compared to other approaches, achieving an accuracy of 91%. This results proves that there is a improved performance in IoT environments. **Experiment (Zero Trust Integration in Federated Learning) -** The Zero Trust framework highlighted effectiveness in addressing simulation attacks in federated learning. This performance showed a significant improvement in accuracy, from 91% without trust filtering to 98% with trust filtering. The results show that the role of Zero Trust principles improves the security and reliability of Federated learning systems.

7 Conclusion and Future Work

In Conclusion, the research goal is a performance comparison of different machine learning algorithms on multi-class IDS in a 5G IoT environment. To meet the objectives, a series of experiments with base classifiers and hybrid models, federated learning, and a Zero Trust mechanism to be integrated into the federated system are performed. The results show that the advanced hybrid models have provided superior performance, proving that the Hybrid Stacking Model achieves the highest accuracy of 99% in both datasets. FL with Stacking and Majority voting model have proven to be the best balance between security and scalability achieving

91% accuracy while effectively handling distributed data. This Zero Trust integration has moved the performance metrics beyond these numbers by dynamically filtering out corrupted clients leading to fine-tuning the reliability and performance of the system from 91% (without trust filtering) to 98% (with trust filtering). This shows how powerful the model becomes when you add advanced machine-learning techniques together with federated learning and zero-trust principles. In Future work, find the most important features using advanced methods to make the model smaller and faster to train without losing accuracy. Use different FL methods that can automatically adjust each client's contribution based on the factors such as network speed and its trustworthiness. In zero trust, test different trust score limits to see how they affect the system, that will help to find a better balance between security and performance

References

Abdalzaher, M.S. *et al.* (2023) 'Toward Secured IoT-Based Smart Systems Using Machine Learning', *IEEE Access*, 11, pp. 20827–20841.

Ahmad, I. *et al.* (2018) 'Overview of 5G Security Challenges and Solutions', *IEEE Communications Standards Magazine*, 2(1), pp. 36–43.

Ahmid, M. and Kazar, O. (2023) 'A Comprehensive Review of the Internet of Things Security', *Journal of Applied Security Research*, 18(3), pp. 289–305.

Al-Juboori, S.A.M. *et al.* (2023) 'Man-in-the-middle and denial of service attacks detection using machine learning algorithms', *Bulletin of Electrical Engineering and Informatics*, 12(1), pp. 418–426.

Alsaedi, A. *et al.* (2020) 'TON-IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems', *IEEE Access*, 8, pp. 165130–165150.

Asad, M. and Otoum, S. (2024) 'Integrative Federated Learning and Zero-Trust Approach for Secure Wireless Communications', *IEEE Wireless Communications*, 31(2), pp. 14–20.

Cao, J. *et al.* (2020) 'A survey on security aspects for 3GPP 5G networks', *IEEE Communications Surveys and Tutorials*, 22(1), pp. 170–195.

Cui, L. *et al.* (2018) 'A survey on application of machine learning for Internet of Things', *International Journal of Machine Learning and Cybernetics*, 9(8), pp. 1399–1417.

Dorogush, A.V., Ershov, V. and Gulin, A. (2018) 'CatBoost: gradient boosting with categorical features support'.

Nour, M. (2023). Implementing Machine Learning to Achieve Dynamic Zero-Trust Intrusion Detection Systems (ZT-IDS) in 5G Based IoT Networks. [online] Proquest.com.

Javeed, D. *et al.* (2024) 'A federated learning-based zero trust intrusion detection system for Internet of Things', *Ad Hoc Networks*, 162, p. 103540.

Lanka, S., Aung Win, T. and Eshan, S. (2021) 'A review on Edge computing and 5G in IOT: Architecture Applications', *Proceedings of the 5th International Conference on Electronics, Communication and Aerospace Technology, ICECA 2021*, pp. 532–536.

Liu, C. *et al.* (2024) 'Dissecting zero trust: research landscape and its implementation in IoT', *Cybersecurity*, 7(1), pp. 1–28.

Mittal, K. and Batra, P.K. (2022) 'Hybrid Machine Learning based Intrusion Detection System for IoT', *3rd IEEE 2022 International Conference on Computing, Communication, and Intelligent Systems, ICCCIS 2022*, pp. 65–69.

Nguyen, V.T. and Beuran, R. (2024) 'FedMSE: Federated learning for IoT network intrusion detection'.

(Mamoun Alazab, Venkatraman, S., Watters, P. and Moutaz Alazab (2011). Zero-day Malware Detection based on Supervised Learning Algorithms of API call Signatures. In Proc. Australasian Data Mining Conference (AusDM 11), [online] 121.

Pinto, A. *et al.* (2023) 'Survey on Intrusion Detection Systems Based on Machine Learning Techniques for the Protection of Critical Infrastructure', *Sensors*, 23(5).

Ramezanpour, K. and Jagannath, J. (2022) 'Intelligent zero trust architecture for 5G/6G networks: Principles, challenges, and the role of machine learning in the context of O-RAN', *Computer Networks*, 217, p. 109358.

Sajid, M. *et al.* (2024) 'Enhancing intrusion detection: a hybrid machine and deep learning approach', *Journal of Cloud Computing*, 13(1).

Seraphim, B.I. *et al.* (2018) 'A survey on machine learning techniques in network intrusion detection system', *2018 4th International Conference on Computing Communication and Automation, ICCCA 2018* [Preprint].

Tariq, A. *et al.* (2024) 'Trustworthy Federated Learning: A Comprehensive Review, Architecture, Key Challenges, and Future Research Prospects', *IEEE Open Journal of the Communications Society*, 5, pp. 4920–4998.

Virat, M.S. *et al.* (2018) 'Security and Privacy Challenges in Internet of Things', *Proceedings of the 2nd International Conference on Trends in Electronics and Informatics, ICOEI 2018*, pp. 454–460.

Wu, R. *et al.* (2023) 'MDIFL: Robust Federated Learning Based on Malicious Detection and Incentives', *Applied Sciences 2023, Vol. 13, Page 2793*, 13(5), p. 2793.

Zhang, T. *et al.* (2021) 'Federated Learning for Internet of Things: A Federated Learning Framework for On-device Anomaly Data Detection'.

ZhangYifei *et al.* (2024) 'A Survey of Trustworthy Federated Learning: Issues, Solutions, and Challenges', *ACM Transactions on Intelligent Systems and Technology* [Preprint].