National College of Ireland

Evaluating the Effectiveness of Neural Networks for Cyber Attack Detection in IoT Ecosystems

MSc Research Project
MSc Cloud Computing

Lyuble Daniel
Student ID: x23249463

School of Computing
National College of Ireland

Supervisor: Dr. Jorge M. Cortés-Mendoza

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……. Lyuble Daniel……

Student ID: …… x23249463……..……

Programme: ….. MSc Cloud Computing…… **Year:** …2024-2025…..

Module: ……Research Project……

Supervisor: …… Dr. Jorge M. Cortés-Mendoza ……

Submission Due Date: ……15 – 09 - 2025……

Project Title: Evaluating the Effectiveness of Neural Networks for Cyber Attack Detection in IoT Ecosystems……

Word Count: …….7027…… **Page Count:** ……21……..……..

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: ……Lyuble Daniel……

Date: ……15-09-2025……

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):	

Evaluating the Effectiveness of Neural Networks for Cyber Attack Detection in IoT Ecosystems

Lyuble Daniel
Student ID: x23249463

Abstract

The introduction of Internet of Things (IoT) devices into the critical infrastructures has resulted in a greatly increased attack surface and the need of an Intrusion Detection Systems that are able to deliver in real time. This research explores the usefulness of Artificial Neural Networks (ANN) and Long Short-Term Memory (LSTM) in identifying various IoT-specific attacks. It compares ANN and LSTM with common machine learning algorithms such as Random Forest (RF), XGBoost, and Support Vector Machine (SVM) using the standard CICIoT2023 dataset like the environment of a real-world traffic in 105 IoT devices. Data preprocessing, class balancing using Synthetic Minority Oversampling Technique (SMOTE) and hyperparameter optimization were used. The evaluation metrics were accuracy, precision, recall, F1- score, Area Under the Curve (AUC) and time to execute. It has been shown that RF provided the best scores with 95 percent accuracy during the test and an almost perfect AUC value of 0.9992, whereas LSTM achieved 77 percent and 0.96 on AUC, which indicates better temporal pattern recognition of this approaches. ANN performed mediocrity with 0.94 AUC and 69 percent accuracy. The results indicate that ANN and LSTM show a great potential in learning the complex behavior of traffic yet, at the same time, the traditional machine learning, and in particular the RF, offers a very useful result in detecting IoT attacks in a manageable amount of computing resources.

1 Introduction

The exponential growth of the Internet of Things (IoT) into a worldwide network of billions of interconnected physical items has revolutionised digital infrastructure across healthcare, industrial systems, smart cities, and critical infrastructures (Asadi et al., 2024). This high degree of interconnection, nevertheless, increases the vulnerability to several cyber threats, especially Distributed Denial of Service (DDoS) assaults as well as spoofing campaigns, reconnaissance activities, brute-force attacks, and newly-emerging malware, like Mirai. With the growing implementation of IoT technologies, the need to implement strong, automated, and smartly adaptive anomaly detection mechanisms gets even stronger (Orman, 2025). In current cybersecurity practice, Machine Learning (ML) and Deep Learning (DL) have taken central positions in the generation of contemporary defense structures since they can recognize complex, rule-based attacks that cannot be tackled by traditional fixed rule-based systems (SOW and Adda, 2024). These limitations can be overcome with DL algorithms, particularly Artificial Neural Networks (ANN) and Long Short-Term Memory (LSTM), which incorporate automatic feature extractor and sequence learning, both of which are essential to the correct identification of the ever-changing attack vectors on a real time basis (Sharma and Bairwa, 2025).

1.1 Problem Statement

Despite the gradual adoption of ML technologies by the Intrusion Detection Systems (IDS), the primary performance bottleneck remains to be the ability to effectively and accurately detect highly advanced IoT-specific attacks (Jamshidi et al., 2025). Most of the traditional ML models that are currently in wide use require a lot of manual feature engineering and have low capabilities to take into account temporal patterns or context-specific behavior that is typical of IoT traffic. DL, on the other hand, with its data-driven designs shows potential to perform better on detection (Al-Shurbaji et al., 2025). However, its high requirements in terms of computations, and complex procedures of training cast serious doubts as to whether it can operate in real-time or scale to the IoT setup.

1.2 Motivation

The quick integration of IoT technology in essential sectors has expanded the attack surface and introduced new vulnerability spots that are challenging to address with conventional security measures (Ogab et al., 2025). Hackers often use unpatched devices, default passwords or poor communication protocols to access IoT systems hence causing major disruption or unauthorized access. At the same time, modern attack techniques have become complex in nature, necessitating the need of defense mechanisms that are equally dynamic and smart (Coston, Plotnizky. and Nojoumian, 2025).

ML algorithms have a lot of potential, but they simply do not generalize well when applied to both rapidly changing attack environments and when feature dependencies are nonlinear and time-varying. The LSTM network and DL techniques in general are well suited to time-dependent data modeling and learning of raw features (Mallid and Ramisetty, 2025). Still, their practical performance is underdeveloped, especially on the datasets like CICIoT2023. This gap drives the need to conduct an in-depth examination that measures the relative detection performance and can guide future IoT solutions to cybersecurity.

1.3 Aim

This research aims to develop a robust and accurate anomaly (attack) detection system for real-world IoT environments using ML and DL algorithms. Acknowledging the critical need for effective security analytics in the face of diverse and large-scale IoT attacks: DDoS, DoS, Reconnaissance, Spoofing, and Mirai, as detailed in the dataset, this research will specifically investigate and compare the efficacy of Random Forest (RF), XGBoost, and Support Vector Machines (SVM) (Katal and Singh, 2021). By utilizing the rich, real-time traffic data from 105 actual IoT devices acting as both attackers and victims, this research will focus on assessing the capabilities of neural networks to automatically learn complex patterns and identify malicious or benign IoT network traffic.

1.4 Research Question

- How do some deep learning techniques like ANN and LSTM compare in performance to traditional machine learning models for detecting diverse attack categories (DDoS, DoS, Reconnaissance, Spoofing, Mirai) within the IoT environment?

1.5 Research Contributions

- A comparative assessment of deep learning models against classical machine learning models for the detection of cyber-attacks in IoT networks.
- The establishment of a standardized methodology for the CICIoT2023 dataset, tackling issues of missing data, class imbalance, and feature translation.
- An evaluation of the models with the standard metrics in the literature like accuracy, adaptability, and computational effectiveness across all models.
- Evaluation of the appropriateness and constraints of deep learning architectures in resource-limited IoT settings.

1.6 Report Structure

The dissertation is organized as follows: Chapter 1 introduces the research problem, objectives, motivation, research questions, and contributions, Chapter 2 reviews and critiques prior studies on IoT security, anomaly detection, and intrusion detection frameworks, Chapter 3 details the research design, data preprocessing, model development, and evaluation metrics, Chapter 4 outlines the system architecture, components, and technical design of the proposed solution, Chapter 5 describes the development process, tools used, and integration of the proposed system, Chapter 6 presents experimental results, case studies, and performance analysis with a critical discussion and Chapter 7 summarizes key findings, addresses limitations, and suggests future research directions.

2 Related Work

2.1 Hybrid Intrusion Detection Techniques for IoT and Industrial IoT (IIoT) Systems

Several studies have advanced IoT/IIoT intrusion detection using ML, DL, and hybrid models. Orman (2025) achieved 99.99% accuracy with an optimized MLP on WUSTL-IIoT-2021, though its generalizability is limited. Imtiaz et al. (2025) used a CNN-spectrogram IDS with >99% accuracy but low adaptability, while Chen et al. (2025) introduced SICNN with real-time detection yet higher false positives. Rahmati (2025) applied RNN with Federated Learning (FL), attaining >98% accuracy but facing latency in large-scale settings. Other works, Ragab et al. (2025), Alzakari et al. (2025), Belachew et al. (2025), Rathnamala et al. (2025), and Bao et al. (2025) reported 99–99.8% accuracy using hybrid CNN-based models, attention mechanisms, or edge-XGBoost, though they suffer from high complexity, resource demands, and limited scalability or real-world validation.

The following related literature applies the CICIoT2023 dataset similar to the proposed research. Jony and Arnob (2024) explored four ML models on the CIC-IoT2023 dataset for IoT attack detection. Decision Tree (DT) and RF showed the highest accuracy (~99.2%), while Logistic Regression (LR) performed the poorest. The study suggests further research into Deep Learning (DL) and unsupervised methods. Alabdulatif et al. (2024) used XGBoost, MLP, and RF to detect DoS/DDoS attacks. With Particle Swarm Optimization (PSO) for feature selection, XGBoost achieved 99.93% accuracy, outperforming traditional methods and enabling a real-time detection system. Mahdi et al. (2025) proposed a hybrid model combining LSTM and Naive Bayes with feature selection to detect DDoS and spoofing attacks. Tested on three datasets, the model achieved up to 99.91% accuracy while reducing false positives. Abebe et al. (2025) applied ML models to classify multiple IoT attacks. DT outperformed others with 98.34% accuracy, showing strong performance over SVM, Bayes, and LR. Al-Ghamdi and Alansari (2025) compared CNN, Bidirectional LSTM (BiLSTM), and hybrid CNN-BiLSTM models on the CICIoT2023 dataset. CNN achieved the best accuracy (98%), proving effective for spatial feature extraction in IoT intrusion detection. Table 1 shows the comparison between various research conducted on the detection of IoT systems.

Table 1: Brief Summary of related works

Article	Main Goal	Technique	Metrics used	Dataset Used	Tools/ Environment
Rathnamala et al. (2025)	Botnet detection	ANN + CNN + BiLSTM + GRU	Accuracy, F1	UNSW-NB15	DL Stack
Bao et al. (2025)	DDoS detection under latency	CNN-mLSTM-KAN + AFSB	Accuracy, Response Time	CICDDoS2019	Edge
Jony and Arnob (2024)	IoT attack detection	DT, RF	Accuracy	CICIoT2023	Python and scikit-learn libraries
Alabdulatif et al. (2024)	Detect DoS/DDoS attacks	XGB	Accuracy	CICIoT2023	Python-based framework
Mahdi et al. (2025)	Detecting DDoS/spoofing and reducing alarms	LSTM and Naive Bayes	Accuracy	CICIoT2023	Python
Abebe et al. (2025)	Detect and classify multiple IoT attacks	DT	Accuracy	CICIoT2023	Python
Al-Ghamdi and Alansari (2025)	IoT attack detection	CNN, CNN-BiLSTM	Accuracy	CICIoT2023	Python and scikit-learn libraries
Proposed Research	***IoT attack detection***	***ANN, LSTM, RF, SVM, XGBoost***	***Precision, Recall, F1***	***CICIoT2023***	***Python***

Recent studies applied various ML and DL models for IoT intrusion detection with high accuracy. Orman (2025) used DT, RF, MLP, and CNN on WUSTL-IIoT-2021 achieving 99.9% accuracy, while Imtiaz et

al. (2025) used CNN + XAI on KDD CUP99 and others with >99%. Chen et al. (2025) achieved high accuracy using SICNN on CICIoT2023, and Rahmati (2025) reached >98% using RNN + FL. Works on CICIoT2023, like Jony and Arnob (2024), Alabdulatif (2024), Mahdi (2025), Abebe (2025), and Al-Ghamdi (2025), reported 94–99.9% accuracy using DT, RF, XGBoost, CNN, and CNN-BiLSTM models. Although these studies achieve strong accuracy, most rely on complex hybrid or deep models needing heavy resources and rarely compare ML and DL approaches together. Few evaluate efficiency (time, resources) or focus on diverse attacks in realistic IoT traffic. This creates a gap that this study addresses by systematically comparing ML (RF, XGBoost, SVM) and DL (ANN, LSTM) on CICIoT2023 for both accuracy and practicality.

2.2 Summary of Existing Literature

The main characteristics of the related work highlight that DL models, including CNN, LSTM, and hybrid variants, generally achieve very high accuracy (often above 99%) by capturing complex spatial and temporal patterns in IoT traffic, but they are resource-intensive, less interpretable, and difficult to scale in real-world deployments. Hybrid approaches such as MHARNN-EGTOCRD and IADCLK further enhance detection through attention mechanisms, optimization algorithms, and adaptive feature selection, showing excellent performance but still requiring validation in practical IoT settings. In contrast, traditional machine learning models like DT, RF, and XGBoost remain strong due to their efficiency, simplicity, and interpretability, consistently achieving accuracies above 98%, although they are less effective in handling sequential dependencies. Federated and edge-based approaches address privacy and latency but face challenges with scalability. Overall, the literature reflects a trade-off between the high accuracy and complexity of DL methods and the practicality and efficiency of classical ML models for IoT anomaly detection.

2.3 Identified Research Gap

Despite extensive research in IoT anomaly detection using ML and DL, key gaps remain. Existing studies often use synthetic datasets and complex models that lack real-world applicability in resource-constrained environments. Few works conduct in-depth evaluation of specific attacks like Mirai, spoofing, or reconnaissance. Moreover, there is limited comparison of traditional ML models (e.g., SVM, RF, XGBoost) against DL models (e.g., ANN, LSTM) in terms of accuracy, efficiency, and interpretability. Most research overlooks real-time traffic and diverse attack types. This study addresses these gaps by using the CICIoT2023 dataset to systematically compare ML and DL models under realistic and heterogeneous IoT conditions.

3 Research Methodology

3.1 Research Design

This project adopts the Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology, which involves problem formulation, dataset selection, preprocessing (including handling missing data and class imbalance with SMOTE), exploratory data analysis, model selection and implementation using both ML and DL approaches, and comprehensive multi-metric evaluation (Shimaoka, Ferreira, and Goldman, 2024). This structured approach ensures that the results are replicable, scalable, and practically relevant to IoT-based intrusion detection systems.

3.2 Dataset Selection and Pre-processing

In this research, the CICIoT2023 dataset provided by the Canadian Institute for Cybersecurity initially contained 65,724 rows and 40 features. During cleaning, 2 rows with null values (one each in the ‘Std’ and ‘Variance’ columns) and 3,574 duplicate rows were removed to ensure data integrity and prevent bias, resulting in 61,558 unique rows. The dataset originally contained 34 fine-grained attack types, which were mapped into 6 main categories: DDoS (44,465 samples), DoS (10,815), Mirai (3,663), Benign (1,471), Spoofing (711), and Reconnaissance (433). To address the severe class imbalance, SMOTE oversampling was applied to equalize each class to 44,465 instances, producing a balanced dataset of 266,790 records with 40 numerical features.

3.3 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) was conducted to gain insights into feature distributions and class balance. Count plots were first used to visualize the distribution of attack categories, confirming the initial imbalance. Box plots were then applied to variables such as average packet size (AVG) to observe spread, which are indicative of abnormal traffic patterns like DDoS and Spoofing (Figure 8). Similarly, violin plots were employed to analyze the distribution of inter-arrival time (IAT) across categories, showing that DDoS and DoS attacks had tightly clustered timings, while Spoofing and Reconnaissance displayed broader spreads, and benign traffic showed balanced distributions (Figure 9). These visualizations validated that distinct statistical patterns exist across different types of attacks, supporting their relevance for classification.

3.4 Model Selection

The models used in the research were selected based on their applicability in terms of IoT intrusion detection and their capability to represent the various attack patterns. RF and XGBoost has been chosen due to its capability of working with high dimensional data and reducing overfitting (Orman, 2025; Belachew et al., 2025). SVM was selected as it has proven effective in simple-to-medium-sized datasets in classification tasks (Kumari, Gaikwad and Chavan, 2025). ANN have been chosen because of their ability to model non-linear relationships (Orman, 2025). The reason behind the use of LSTM was their ability to model sequential behavior of network traffic (Mahdi et al., 2025). The strengths of all these models are its non-linear learning capabilities, time-dependent behaviors and interpretability. Nevertheless, some models make computations more complex and prone to overfitting. Still, the selected methodology is sufficient to answer the research question and have a scale-able solution in future smart IoT environments.

3.5 Machine Learning Models

3.5.1 Support Vector Machine (SVM)

SVM is specially noted for its capability to identify the best hyperplane as shown in Figure1, that gives the maximum margin of separation (Kumari, Gaikwad and Chavan, 2025).

$$\min_{\omega,b} \frac{1}{2}\|\omega\|^2 \qquad subject\ to\ y_i[\omega^T \varphi[x_i] + b] \geq 1$$

w is the weight vector perpendicular to the hyperplane, b is the bias term, $y_i \in \{-1, +1\}$ is the class label of the ith sample, x_i is the feature vector of the ith sample,
$\varphi(x_i)$ is the mapping function projecting inputs into higher-dimensional space and the objective is to minimise $\|w\|^2$ while ensuring each sample lies on the correct side of the hyperplane.

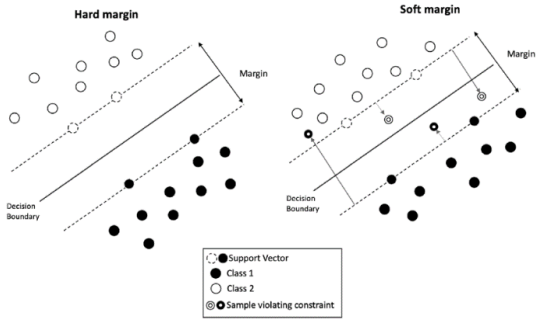

Figure 1: SVM (Prinzi et al., 2024)

Figure 1 compares hard margin and soft margin SVMs.

- Hard margin SVM perfectly separates the classes with no misclassification, assuming linearly separable data.
- Soft margin SVM allows some violations (misclassified or borderline points) to improve generalization on non-separable data.

3.5.2 Random Forest (RF)

RF is an ensemble classification algorithm which constructs a series of decision trees as shown in Figure 2 and determines the class based on the mode of predictions from these trees (Orman, 2025).

$$\hat{y} = majority\_vote\{h_t(x)\}_{t=1}^{T}$$

$\hat{y}$ = mode($\{h_1(x), h_2(x), \ldots, h_t(x)\}$) is the final prediction,
$h_t(x)$ is the prediction from the tth decision tree,
T is the total number of trees,
$majority\_vote$ denotes the class that appears most frequently among the predictions

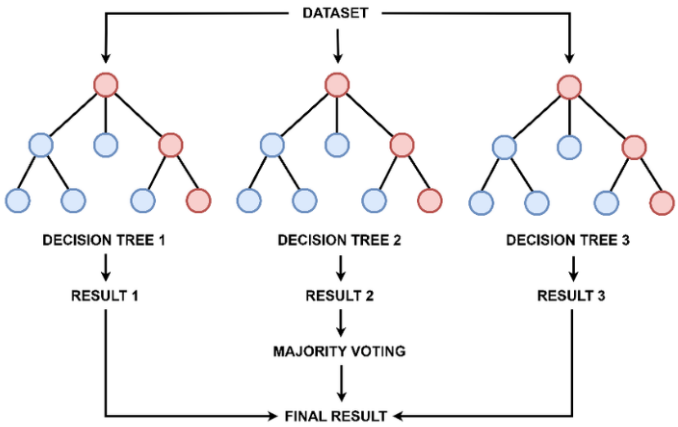

Figure 2: RF (Nur, Wijaya and Wulandari, 2024)

3.5.3 XGBoost (Extreme Gradient Boosting)

XGBoost is a scalable, regularized boosting algorithm, uses gradient descent to minimize the loss function by sequentially adding weak learners in an ensemble (Belachew et al., 2025).

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \varepsilon F$$

$\hat{y}_i = \Sigma_k f_k(x_i)$ is the predicted output for instance i,
K is the total number of trees,
$f_k \in F$ represents the kth decision tree from the space F of all regression trees,
and each new tree is added to reduce the residual errors of the previous ensemble.

Each tree f_k is added sequentially to reduce the error from the previous ensemble as shown in Figure 3, gradually improving the model's accuracy.

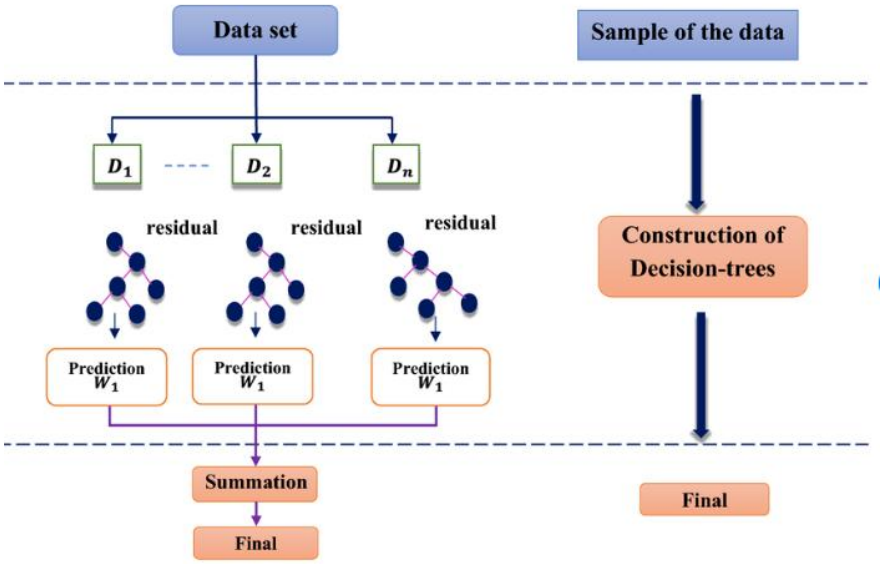

Figure 3: XGBoost (Das et al., 2024)

3.6 Artificial Neural Networks (ANN)

ANN is a feedforward network whose layers are fully connected as shown in Figure 5, in which neurons use weighted sums followed by non-linear activations to learn abstract representations of input data (Oyinloye, Arowolo and Prasad, 2025).

$$Predicted\ Output\ \hat{y} = \varphi\ (w_n.\varphi\left[w_{n_{-1}} \dots \cdots \varphi(w_i \cdot x + b_i) + b_{n-1}\right] + b_n]$$

x is the input feature vector,
W_i is the weight matrix of the ith layer,
b_i is the bias vector of the ith layer,
f is the non-linear activation function,
and the network computes $a_i = f(W_i x + b_i)$ through fully connected layers.
φ- activation function.

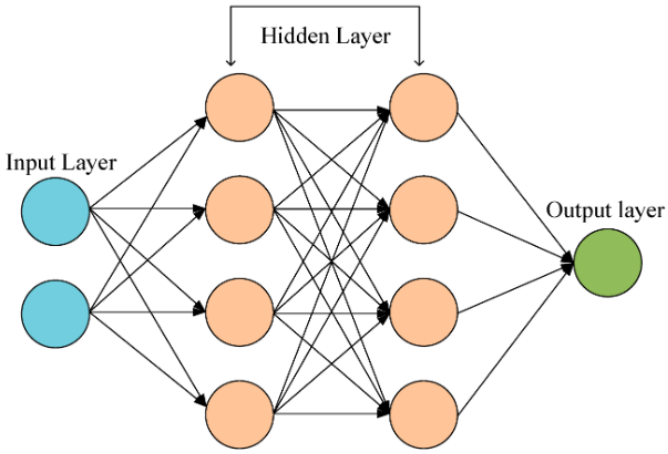

Figure 5: ANN (Khan et al., 2025)

3.7 Long Short-Term Memory (LSTM)

LSTM contains memory cells and gating units in order to remember information over time steps so it can be applied to sequential data as shown in Figure 4 (Dash et al., 2025).

$$f_t = \sigma\left(w_f \cdot [h_{t-1}, x_t] + b_f\right)$$

$$i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\tilde{C_t} = tanh\ (w_c \cdot [h_{t-1}, x_t] + b_c)$$
$$C_t = f_t \Theta C_{t-1} + i_t \Theta \tilde{C_t}$$
$$o_t = \sigma(w_o \cdot [h_{t-1}, x_t] + b_o)$$
$$h_t = o_t \Theta\ tan\ tan\, h\,(c_t)$$

Where f_t, i_t, o_t are the forget, input, and output gates, $\hat{c}_t$ is the candidate cell state, c_t is the cell state, h_t is the hidden state, x_t is the input at time t, σ is the sigmoid function, and tanh is the hyperbolic tangent activation.

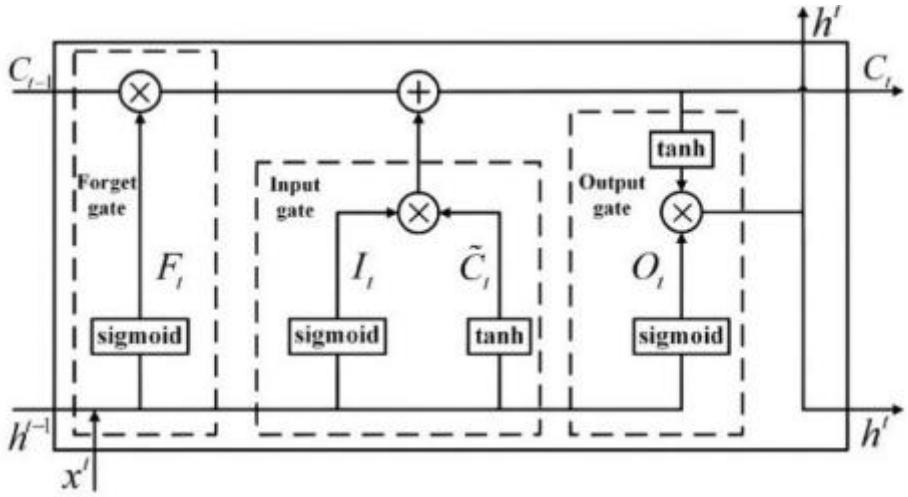

Figure 4: LSTM (Dash et al., 2025)

3.8 Evaluation Metrics

In order to evaluate the model's ability to detect IoT cyberattacks, Accuracy, Precision, Recall, F1-Score, Area Under the Curve (AUC), Confusion Matrix, and Execution Time were used. Accuracy provides a general measure of correctness but can be misleading when the datasets are imbalanced and therefore Precision and Recall can be used in order to assess the relevance and completeness of the attacks detected (Nawaz et al., 2025). F1-Score averages these two, so it is appropriate when classes are imbalanced. AUC represents the capability of the model to differentiate attack and benign traffic in thresholds. Confusion Matrix provides the detailed picture of both correct and wrong predictions per each of the classes. Execution Time is also presented in order to estimate the computing performance of every model (Belachew et al., 2025) A combination of these metrics allows the evaluation to look at both the detection effectiveness, and practical deployment limits.

Accuracy (Acc) = $\frac{TP+TN}{TP+FP+TN+FN}$

Precision (P) = $\frac{TP}{TP+FP}$

Recall(R) = $\frac{TP}{TP+FN}$

F1 Score = $2*\frac{P*R}{P+R}$

$$AUC = \sum_{i=1}^{n}(TPR_i + TPR_{i-1}) * (FPR_i + FPR_{i-1})$$

$$TPR = \frac{TP}{TP + FN} \qquad FPR = \frac{FP}{FP + TN}$$

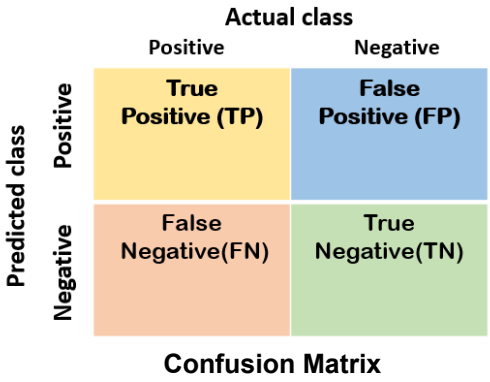

Figure 6: Evaluation Metrics Formula and Confusion Matrix (Original Source)

4 Design Specification

4.1 Project Architecture

The workflow of this project follows the CRISP-DM methodology, beginning with problem understanding, where the research objective is defined as detecting IoT-based cyberattacks using ML and DL models.

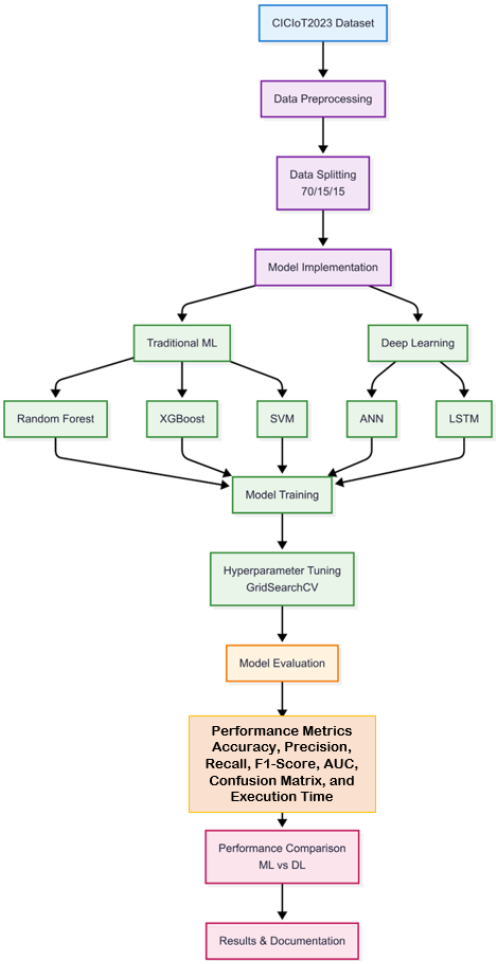

Figure 7: Project Workflow (Original source)

In the data understanding phase, the CICIoT2023 dataset is explored to identify the different attack types and categories. The data preparation phase involves preprocessing steps such as cleaning missing values, removing duplicates, encoding categorical features, and addressing class imbalance using SMOTE, followed by splitting the dataset into training, validation, and testing sets in a 70:15:15 ratio. In the modeling phase, RF, XGBoost, SVM, ANN and LSTM are implemented and trained, with hyperparameter tuning carried out using GridSearchCV. The evaluation phase then measures model performance using accuracy, precision, recall, F1-score, AUC, confusion matrix, and execution time, enabling a direct comparison between ML and DL approaches.

4.2 ML Model Hyperparameter Tuning

RF, XGBoost, and SVM—were implemented. GridSearchCV with 2-fold cross-validation was used for hyperparameter tuning, systematically exploring parameter combinations to identify the optimal configuration for accurate and efficient cyberattack classification. A 2-fold cross-validation was selected to reduce computational cost and training time on the large, high-dimensional CICIoT2023 dataset,

providing an efficient balance between accuracy estimation and runtime (Burman, 1989). The parameters tested are listed in Table 2.

Table 2: ML parameters for tuning

Algorithms	Hyperparameters
RF	n_estimators: [50, 100, 200, 300], max_depth: [10, 20, 5, 30], min_samples_split: [5, 10, 2, 5], min_samples_leaf: [1, 2, 3, 4, 5], max_features: ['sqrt', 'log2', None]
XGBoost	n_estimators: [50, 100, 200, 300], learning_rate: [0.03, 0.02, 0.3, 0.2], max_depth: [2, 3, 4, 6, 8], subsample: [0.7, 0.6, 0.4, 0.2], colsample_bytree: [0.7, 0.6, 0.5, 0.4], gamma: [0, 0.1, 0.2, 0.3], reg_alpha: [0, 0.1, 0.2, 0.3], reg_lambda: [1, 2, 3, 4]
SVM	C: [0.1, 0.2, 0.3], kernel: ['rbf', 'poly', 'sigmoid', 'linear'], gamma: ['scale', 'auto'], max_iter:[500, 300, 100], tol: [1e-2, 1e-1, 1e-3]

4.3 ANN and LSTM Architectures

The LSTM model is composed of a single LSTM layer with 64 units and a dropout rate of 0.3, followed by a dense layer of 32 units with a 0.2 dropout. To preserve computational efficiency and prevent overfitting, this minimal architecture is selected to capture temporal dependencies in IoT traffic data. A softmax-activated output layer is implemented for multiclass classification, employing the sparse categorical cross entropy loss.

The ANN architecture is comprised of two hidden dense layers: the first contains 64 units and a 0.3 dropout, while the second has 32 units and a 0.2 dropout. Both layers were subjected to L2 regularisation (0.01) in order to enhance generalisation and mitigate overfitting. The architecture is amenable to learning from structured input features, and a softmax output layer is responsible for multiclass prediction tasks.

A staged hyperparameter optimisation approach is implemented for both LSTM and ANN models. Initially, a variety of optimisers (including Adam, Adamax, and RMSprop) were evaluated. A variety of learning rates (0.001 to 0.004) were employed to fine-tune the optimiser that demonstrated the highest performance. Once the optimal learning rate was determined, activation functions (ReLU, tanh) were assessed. Ultimately, the optimal convergence point was determined by investigating various epoch configurations (50–200). By segregating and refining the effect of each hyperparameter, this sequential tuning strategy guaranteed optimal performance across both architectures, thereby enhancing model generalisation and stability.

Table 3 provides the settings or configurations of both the models.

Table 3: Parameters of ANN and LSTM deep learning Architectures

Parameter	ANN (Value/Setting)	LSTM (Value/Setting)
Hidden Layers	First layer: 64 units - with dropout of 0.3 Second Layer: 32 units - with dropout of 0.2	One LSTM layer: 64 units – with dropout of 0.3 One Dense layer: 32 units – with dropout of 0.2
Output Layer	6 units, softmax	6 units, softmax
Kernel Regularizer	l2(0.01) for both hidden layers	Not applied
Loss Function	sparse_categorical_crossentropy	sparse_categorical_crossentropy
Batch Size	64	64

Optimizer Functions	Adam, Adamax, RMSprop	Adam, Adamax, RMSprop
Activation Functions	Relu, tanh	Relu, tanh
Learning Rate	0.001, 0.002, 0.003, 0.004	0.001, 0.002, 0.003, 0.004
Epochs	50, 100, 150, 200	50, 100, 150, 200

5 Implementation

5.1 Dataset Preparation and Cleaning

The collected dataset contains 65,724 rows and 40 columns in which 34 unique attack types are present in the Label (Output) column. The attack types are mapped into broader categories, and the mapping is applied to the label column, which resulted in a total of 14 categories. For this experiment the labels other than 'DDOS', 'DOS', 'RECON', 'SPOOFING', 'MIRAI', 'BENIGN' are dropped. The only main attack categories with sufficient data (DDOS, DOS, MIRAI, SPOOFING, RECON, BENIGN) were retained to guarantee robust learning and generalisation. Exclusion of rare classes was necessary due to their exceptionally low sample sizes, which could potentially result in class imbalances and a decrease in model performance.
The dataset was cleaned through the following steps:

- Removed **2 rows with null values** (in ‘Std’ and ‘Variance’ columns).
- Removed **3,574 duplicate rows** to prevent redundancy and bias.
- Mapped **34 fine-grained attack types** into **6 main categories** (DDoS, DoS, Mirai, Spoofing, Reconnaissance, Benign).
- Retained only these 6 categories, dropping rare classes with very few samples.
- Converted categorical columns to numeric format using LabelEncoder.

After cleaning, the dataset contained **61,558 unique rows and 40 features**, ready for encoding and oversampling.

5.2 Data Encoding, Oversampling and Splitting

The initial dataset was significantly imbalanced, with reconnaissance and benign attacks being under-represented. Using the LabelEncoder, attack categories were numerically encoded. SMOTE was implemented to rectify the imbalance, resulting in an equalization of all class samples at 44,465 each. This guaranteed a balanced dataset of 266,790 instances (with 44,465 per each label) and 40 features, thereby enhancing the accuracy and impartiality of model training.

Table 4: Attack Label Encoding and Class Distribution

Labels	Encoded Labels	Original Value count
DDOS	1	44465
DOS	2	10815
MIRAI	3	3663
BENIGN	0	1471
SPOOFING	5	711
RECON	4	433

The cleaned data is splitted into three subsets- 70% for training, 15% for validation and 15% for testing:

- Training data = 186753 instances
- Validation data = 40018 instances
- Testing data = 40019 instances

5.3 Exploratory Data Analysis

Box plot here shows how AVG varies across different attack categories. Figure 8 shows the distribution of AVG across attack categories, highlighting distinct patterns where benign traffic exhibits higher variability, while attack types such as DDoS, DoS, and Mirai display more uniform packet sizes.
In cybersecurity datasets, extreme values often correspond to abnormal but real attack behaviors, such as unusually small packets in DoS/DDoS floods or irregularly large packets in spoofing and reconnaissance traffic. Removing these data points would risk eliminating attack-specific characteristics that are crucial for effective classification. Moreover, RF and XGBoost are relatively robust, meaning that their presence does not distort learning but instead helps the models capture the full distribution of both benign and malicious traffic patterns. Therefore, the majority of observations are clustered around the mean, while some instances deviate significantly. In intrusion detection, such diversity reflects realistic network behavior where both normal and malicious traffic can vary widely. Retaining these diverse values during training exposes the model to rare but critical attack patterns, thereby improving its robustness and generalization to future, unseen traffic scenarios.

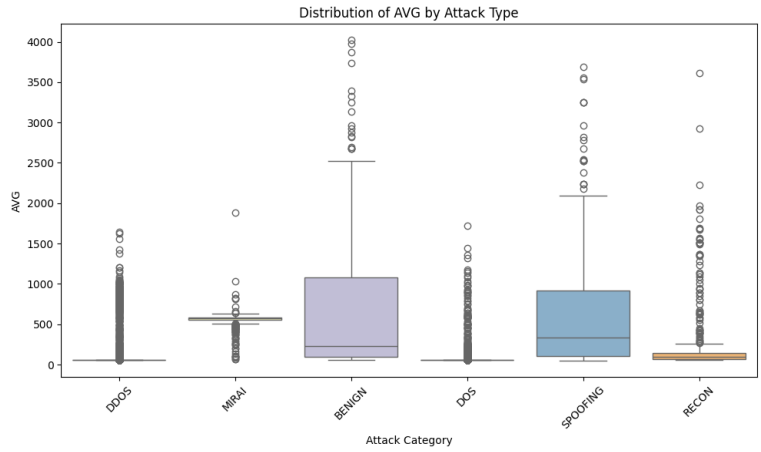

Figure 8: Box plot showing AVG distribution across different attack categories

Violin plot here visualizes IAT distribution for each IoT attack category, highlighting data spread, density, and median inter-arrival time.

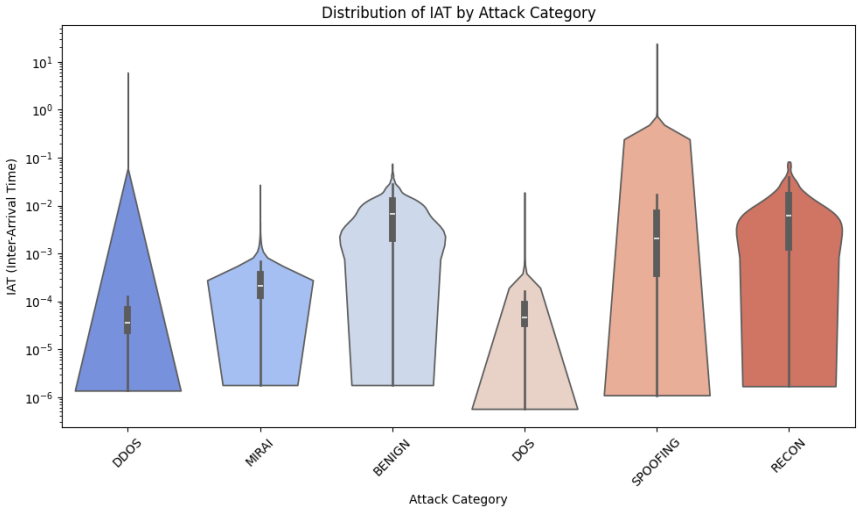

Figure 9: Violin plot showing IAT distribution across different attack categories

The violin plot demonstrates that DDoS and DoS attacks have inter-arrival times that are lower and densely packed, which suggests that the traffic is rapid and irregular. Spoofing and Recon have broader distributions, which suggests that the traffic is slower and more irregular. Benign traffic has a moderate spread and balanced timing.

The violin plot in Figure 9 illustrates the distribution of IAT across attack categories by showing both spread and data density. DDoS and DoS traffic exhibit tightly clustered, lower IAT values, reflecting rapid and bursty packet flows. In contrast, Spoofing and Reconnaissance show broader distributions, indicating more irregular and slower traffic patterns, while Benign traffic displays a moderate spread with balanced timing. This visualization demonstrates that IAT captures distinct temporal characteristics useful for differentiating between normal and malicious IoT traffic. EDA showed clear differences between attack types. Box plots revealed consistent packet sizes for DDoS, DoS, and Mirai, while benign traffic was more variable. Violin plots showed dense, rapid IATs for DDoS/DoS and broader, slower patterns for Spoofing/Recon, with benign traffic balanced. These patterns support using RF/XGBoost for feature-based detection and LSTM for capturing temporal behavior.

6 Evaluation

The hyperparameters that maximize the efficiency of the ML models were identified through the successful execution of hyperparameter tuning using GridSearchCV. Parameter configurations that were most effective in the development of the final cyberattack classification models are summarized in Table 5.

Table 5: Best hyperparameters of the ML algorithms for the testing dataset

ML	Hyperparameters	Accuracy	MSE
RF	max_depth: 30, max_features: log2, min_samples_leaf: 1, min_samples_split: 2, n_estimators: 200	0.9460	0.0093
XGB	colsample_bytree: 0.7, gamma: 0, learning_rate: 0.3, max_depth: 8, n_estimators: 300, reg_alpha: 0.3, reg_lambda: 4, subsample: 0.6	0.9426	0.0137
SVM	C: 0.3, gamma: scale, kernel: linear, max_iter: 500, tol: 0.01	0.3821	0.1002

The following table illustrates the performance assessment of three machine learning models. Random Forest consistently generated the highest scores, achieving 0.9819 accuracy, 0.9823 precision and 0.9819 F1 score on the training set, and 0.95 accuracy on the validation and testing sets. Its execution time ranged from 0.9 to 4.3 seconds, and its AUC was 0.9992 during testing.

Table 6: Results of RF, XGBoost and SVM in cyber-attack detection

Model	Stage	Accuracy	Precision	F1 score	AUC	Time (s)
RF	Train	0.9819	0.9823	0.9819	0.9992	4.3095
	Val.	0.9540	0.9542	0.9538	0.9994	0.9086
	Test	0.9530	0.9538	0.9534	0.9992	0.9074
XGBoost	Train	0.9522	0.9523	0.9521	0.9981	0.1642
	Val.	0.9457	0.9457	0.9455	0.9982	0.0385

	Test	0.9435	0.9440	0.9438	0.9981	0.0384
SVM	Train	0.1505	0.1505	0.1505	0.5012	14.5962
	Val.	0.1490	0.1490	0.1490	0.4995	4.3989
	Test	0.1481	0.1481	0.1481	0.5006	3.8223

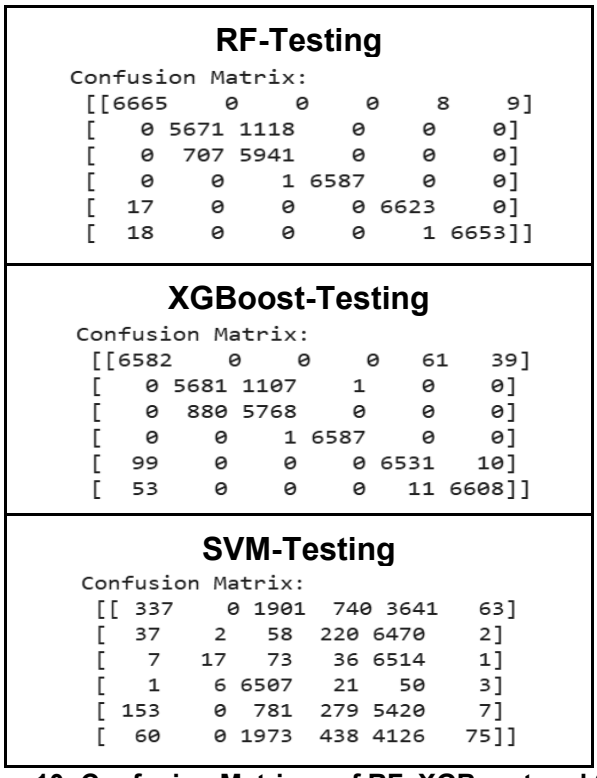

Figure 10: Confusion Matrices of RF, XGBoost and SVM

XGBoost also demonstrated satisfactory performance, with an AUC of 0.9982 and F1 scores ranging from 0.94 to 0.95. The SVM demonstrated subpar performance, with an AUC of approximately 0.50 and an accuracy of only 0.15. In general, Random Forest is recognised as the most effective model for detecting cyber-attacks. In the beginning, three optimizers—Adam, Adamax, and RMSprop—were examined with ReLU activation, 100 epochs, and a learning rate of 0.001. With the Adamax-relu-100-0.001 configuration, Adamax became the most effective optimiser, achieving 0.69 testing accuracy and 0.93 AUC across all stages. Table 7 provides the performance evaluation of ANN tested with varying optimizers.

Table 7: Results of ANN with different optimizers

Best Model Configuration	Stage	Accuracy	Precision	F1 score	AUC	Time (s)
Adamax-relu-100-0.001	**Train**	**0.6964**	**0.7038**	**0.6933**	**0.9343**	**3.6550**
	Val.	**0.6994**	**0.7066**	**0.6961**	**0.9346**	**0.8165**

	Test	**0.6921**	**0.6996**	**0.6901**	**0.9335**	**0.8144**

In the second phase the learning rate were tested using the best performed Adamax optimiser. The configuration Adamax-relu-100-0.002 demonstrated consistently strong performance achieving an AUC during testing of 0.9337 and an accuracy of 0.68 across all three stages.
Table 8 provides the performance evaluation of ANN tested with varying learning rate.

Table 8: Results of ANN with different learning rate

Best Model Configuration	Stage	Accuracy	Precision	F1 score	AUC	Time (s)
Adamax-relu-100-0.002	**Train**	**0.6807**	**0.6993**	**0.6755**	**0.9340**	**3.7042**
	Val.	**0.6819**	**0.7004**	**0.6769**	**0.9341**	**0.8194**
	Test	**0.6787**	**0.6977**	**0.6748**	**0.9337**	**0.8166**

In the third phase the activation functions were tested using the best performed Adamax optimizer and 0.002 learning rate. ReLU consistently outperformed tanh.
Table 9 provides the performance evaluation of ANN tested with varying activation functions.

Table 9: Results of ANN with different activation function

Best Model Configuration	Stage	Accuracy	Precision	F1 score	AUC	Time (s)
Adamax-relu-100-0.002	**Train**	**0.6461**	**0.6602**	**0.6394**	**0.9277**	**3.6592**
	Val.	**0.6452**	**0.6570**	**0.6369**	**0.9279**	**0.7923**
	Test	**0.6424**	**0.6573**	**0.6358**	**0.9268**	**0.8275**

In the fourth phase the epochs were tested using the best performed Adamax optimizer, relu activation function and 0.002 learning rate.
Table 10 provides the performance evaluation of ANN tested with varying epochs.

Table 10: Results of ANN with different epochs

Best Model Configuration	Stage	Accuracy	Precision	F1 score	AUC	Time (s)
Adamax-relu-100-0.002	**Train**	**0.6908**	**0.7220**	**0.6798**	**0.9409**	**3.6443**
	Val.	**0.6927**	**0.7233**	**0.6819**	**0.9413**	**0.7976**
	Test	**0.6884**	**0.7198**	**0.6789**	**0.9401**	**0.8221**

In general, the configurations that demonstrated the most efficacy for cyberattack classification using ANN were Adamax-relu-100-0.001 and Adamax-relu-100-0.002.

Adamax outperformed Adam and RMSprop, obtaining the highest results early on (0.70 accuracy and 0.9429 AUC) among the three optimisers that were initially tested. The findings resulted in additional adjustment using Adamax with varying learning rates and epochs. Table 11 provides the performance evaluation of LSTM tested with varying optimizers.

Table 11: Results of LSTM with different optimizers

Configurations	Stage	Accuracy	Precision	F1 score	AUC	Time (s)
Adamax-relu-100-0.001	Train	0.7019	0.7293	0.6960	0.9439	4.4455
	Val.	0.7040	0.7300	0.6975	0.9441	0.9672
	Test	0.6990	0.7269	0.6941	0.9429	0.9548

In the second phase the learning rate were tested using the best performed Adamax optimiser. Table 12 provides the performance evaluation of LSTM tested with varying learning rate where 0.001 performed well.

Table 12: Results of LSTM with different learning rate

Configurations	Stage	Accuracy	Precision	F1 score	AUC	Time (s)
Adamax-relu-100-0.001	Train	0.7506	0.7787	0.7497	0.9565	4.3052
	Val.	0.7539	0.7815	0.7531	0.9569	0.9408
	Test	0.7467	0.7740	0.7466	0.9551	0.9647

Table 13 provides the performance evaluation of LSTM tested with varying activation functions where relu activation function with learning rate 0.001 and Adamax optimizer performed well.

Table 13: Results of LSTM with different activation function

Configurations	Stage	Accuracy	Precision	F1 score	AUC	Time (s)
Adamax-relu-100-0.001	Train	0.7464	0.7816	0.7430	0.9528	4.3393
	Val.	0.7494	0.7843	0.7464	0.9534	0.9525
	Test	0.7427	0.7782	0.7404	0.9519	0.9377

Table 14 provides the performance evaluation of LSTM tested with varying epochs where 200 epochs performed well.

Table 14: Results of LSTM with different epochs

Configurations	Stage	Accuracy	Precision	F1 score	AUC	Time (s)
Adamax-relu-200-0.001	Train	0.7756	0.8050	0.7736	0.9636	4.7786
	Val.	0.7780	0.8060	0.7760	0.9642	0.9726
	Test	0.7713	0.8005	0.7699	0.9629	0.9548

Adamax-relu-200-0.001 demonstrated the optimal configuration, achieving 0.78 accuracy and 0.9636 AUC during training and sustaining consistent performance throughout the validation and testing stages.

6.1 Discussion

Experimental results for RF, SVM, XGBoost, ANN, and LSTM models offer a comprehensive assessment of their efficacy in the classification of cyberattacks. RF exhibited the highest accuracy, precision, recall, and F1 score among traditional ML models (all 0.98 on training and 0.95 on validation and testing). Additionally, it exhibited an exceptional AUC of 0.9992 and a rapid execution time. The F1 scores of XGBoost were approximately 0.94–0.95, and the AUC was 0.9982, indicating strong and reliable performance, albeit slightly lower. Conversely, the SVM exhibited subpar performance, with an AUC of approximately 0.50 and an accuracy of only 0.15, suggesting that it is not appropriate for this research. In the deep learning domain, ANN models that utilised the Adamax optimiser outperformed other configurations. In particular, Adamax-relu-100-0.001 and Adamax-relu-100-0.002 consistently attained AUCs of 0.94 and an accuracy range of 0.69–0.70 across all stages. ReLU has been found to be more effective than tanh after additional refining of the activation functions and epochs. LSTM models achieved even superior results, with Adamax-Relu 200-0.001 achieving 0.78 training accuracy and 0.96 AUC, while maintaining strong validation and testing scores. In the LSTM model, the Adamax-relu-200-0.001 is the optimal configuration for cyberattack detection, as it exhibits exceptional accuracy, stability, and robustness in all metrics. In detecting a variety of IoT attack categories, including DDoS, spoofing, and Mirai, ANN and LSTM, exhibited a high level of capability, obtaining a 77% testing accuracy. Nevertheless, traditional ML models such as RF outperformed them, achieving a testing accuracy of 95%. Consequently, RF is the most effective overall model for detecting cyberattacks.

Despite their capacity to learn complex temporal patterns, the deep learning models (ANN and LSTM) underperformed compared to traditional ML models. Their lower accuracy and longer execution times indicate that they require larger datasets, extensive tuning, and more computational resources to generalise effectively. This suggests that DL approaches may be less practical for resource-constrained IoT environments without further optimisation.

6.1.1 Comparison with Previous Research

The proposed research achieved 95% accuracy with RF and 94% accuracy with XGBoost, in contrast to prior studies that utilised the same dataset and achieved over 99% accuracy with models such as XGBoost and Decision Trees (DT). Although slightly lower, these results are still competitive, particularly in view of the balanced evaluation and model diversity that were investigated.

Table 15: Comparison of Model Accuracy with Existing Works

Article	Accuracy
Jony and Arnob (2024)	DT: 99.19%, RF: 99.16%
Alabdulatif et al. (2024)	XGB: 99.93%
Mahdi et al. (2025)	hybrid cascaded LSTM + Naive Bayes ensemble: 99.88%
Abebe et al. (2025)	DT: 98.34%

Al-Ghamdi and Alansari (2025)	CNN: 98%, CNN-BiLSTM: 94%, BiLSTM: 85%
Proposed Research	***RF: 95%, XGB: 94%, LSTM: 77%, ANN: 69%***

7 Conclusion and Future Work

This research developed and evaluated traditional ML (RF, XGBoost, SVM) and DL (ANN, LSTM) models for IoT cyberattack detection using the CICIoT2023 dataset. After preprocessing and hyperparameter tuning, models were assessed on accuracy, precision, recall, F1-score, AUC, and execution time. RF performed best overall (95% accuracy, 0.9992 AUC) with strong efficiency, while LSTM (Adamax–ReLU–200–0.001) achieved the top DL performance (77% accuracy, 0.96 AUC) by capturing temporal patterns. ANN showed moderate results (70% accuracy, 0.94 AUC). Although DL models learned complex patterns, RF remained more accurate and efficient. Limitations include reliance on static data, no evaluation of evolving attacks or generalisation, and no deployment or explainability analysis. Future work should explore lightweight DL architectures, real-time/online learning, FL for privacy, and integrating explainable AI (XAI) for interpretability.

References

Abebe, A., Gebeyehu, S. and Alem, A. (2025). Artificial intelligence model for internet of things attack detection using machine learning algorithms. *F1000Research*, *14*, p.230.

Alabdulatif, A., Thilakarathne, N.N. and Aashiq, M. (2024). Machine Learning Enabled Novel Real-Time IoT Targeted DoS/DDoS Cyber Attack Detection System. *Computers, Materials & Continua*, *80*(3).

Al-Ghamdi, A.M. and Alansari, M.M. (2025). Enhancing IoT Security: A Comparative Study of CNN and RNN-Based Anomaly Detection Using the CICIoT2023 Dataset. *IAENG International Journal of Computer Science*, *52*(5).

Al-Shurbaji, T., Anbar, M., Manickam, S., Hasbullah, I.H., ALfriehate, N., Alabsi, B.A., Alzighaibi, A.R. and Hashim, H. (2025). Deep Learning-Based Intrusion Detection System For Detecting IoT Botnet Attacks: A Review. *IEEE Access*.

Alzakari, S.A., Aljebreen, M., Ahmad, N., Alhashmi, A.A., Alahmari, S., Alrusaini, O., Al-Sharafi, A.M. and Almukadi, W.S. (2025). An intelligent ransomware based cyberthreat detection model using multi head attention-based recurrent neural networks with optimization algorithm in IoT environment. *Scientific Reports*, *15*(1), p.8259.

Asadi, M., Jamali, M.A.J., Heidari, A. and Navimipour, N.J. (2024). Botnets unveiled: A comprehensive survey on evolving threats and defense strategies. *Transactions on Emerging Telecommunications Technologies*, *35*(11), p.e5056.

Bao, X., Chen, E. and Tu, X. (2025). Hybrid Deep Neural Network-Based Detection of DDoS Attacks in Software-Defined IIoT Networks.

Belachew, H.M., Beyene, M.Y., Desta, A.B., Alemu, B.T., Musa, S.S. and Muhammed, A.J. (2025). Design a Robust DDoS Attack Detection and Mitigation Scheme in SDN-Edge-IoT by Leveraging Machine Learning. *IEEE Access*.

Burman, P. (1989). A comparative study of ordinary cross-validation, v-fold cross-validation and the repeated learning-testing methods. *Biometrika*, *76*(3), pp.503-514.

Chen, H., Wang, Z., Yang, S., Luo, X., He, D. and Chan, S. (2025). Intrusion detection using synaptic intelligent convolutional neural networks for dynamic Internet of Things environments. *Alexandria Engineering Journal*, *111*, pp.78-91.

Coston, I., Plotnizky, E. and Nojoumian, M. (2025). Comprehensive Study of IoT Vulnerabilities and Countermeasures. *Applied Sciences*, *15*(6), p.3036.

Das, P., Kashem, A., Rahat, J.U. and Karim, R. (2024). A comparative study of ensemble machine learning models for compressive strength prediction in recycled aggregate concrete and parametric analysis. Multiscale and Multidisciplinary Modeling, *Experiments and Design, 7(4),* pp.3457-3482.

Dash, N., Chakravarty, S., Rath, A.K., Giri, N.C., AboRas, K.M. and Gowtham, N. (2025). An optimized LSTM-based deep learning model for anomaly network intrusion detection. *Scientific Reports, 15(1),* p.1554.

Imtiaz, N., Wahid, A., Abideen, S.Z.U., Kamal, M.M., Sehito, N., Khan, S., Virdee, B.S., Kouhalvandi, L. and Alibakhshikenari, M. (2025), January. A deep learning-based approach for the detection of

various Internet of Things intrusion attacks through optical networks. In *Photonics* (Vol. 12, No. 35, pp. 1-39). MDPI.
Jamshidi, S., Nikanjam, A., Wazed, N.K. and Khomh, F. (2025). Leveraging Machine Learning Techniques in Intrusion Detection Systems for Internet of Things. *arXiv preprint arXiv:2504.07220*.
Jony, A.I. and Arnob, A.K.B. (2024). Securing the internet of things: evaluating machine learning algorithms for detecting IoT cyberattacks using cic-iot2023 dataset. *International Journal of Information Technology and Computer Science*, *16*(4), pp.56-65.
Katal, A. and Singh, N. (2021). Artificial neural network: models, applications, and challenges. *Innovative Trends in Computational Intelligence,* pp.235-257.
Khan, S., Rayhan, S.B., Rahman, M.M., Sultana, J. and Varga, G. (2025). Optimized ANN Model for Predicting Buckling Strength of Metallic Aerospace Panels Under Compressive Loading. Metals, 15(6), p.666.
Kumari, M., Gaikwad, M. and Chavan, S.A. (2025). A secure IoT-edge architecture with data-driven AI techniques for early detection of cyber threats in healthcare. *Discover Internet of Things, 5(1),* p.54.
Mahdi, Z.S., Zaki, R.M. and Alzubaidi, L. (2025). Advanced hybrid techniques for cyberattack detection and defense in IoT networks. *Security and Privacy*, *8*(2), p.e471.
Mallidi, S.K.R. and Ramisetty, R.R. (2025). Advancements in training and deployment strategies for AI-based intrusion detection systems in IoT: a systematic literature review. *Discover Internet of Things*, *5*(1), p.8.
Nawaz, M., Tahira, S., Shah, D., Ali, S. and Tahir, M. (2025). Lightweight machine learning framework for efficient DDoS attack detection in IoT networks. *Scientific Reports, 15(1),* p.24961.
Nur, Z.K., Wijaya, R. and Wulandari, G.S. (2024). Optimizing Emotion Recognition with Wearable Sensor Data: Unveiling Patterns in Body Movements and Heart Rate through Random Forest Hyperparameter Tuning. arXiv preprint arXiv:2408.03958.
Ogab, M., Zaidi, S., Bourouis, A. and Calafate, C.T. (2025). Machine Learning-Based Intrusion Detection Systems for the Internet of Drones: A Systematic Literature Review. *IEEE Access*.
Orman, A. (2025). Cyberattack detection systems in industrial internet of things (IIoT) networks in big data environments. *Applied Sciences*, *15*(6), p.3121.
Oyinloye, T.S., Arowolo, M.O. and Prasad, R. (2025). Enhancing cyber threat detection with an improved artificial neural network model. *Data Science and Management, 8(1),* pp.107-115.
Prinzi, F., Currieri, T., Gaglio, S. and Vitabile, S. (2024). Shallow and deep learning classifiers in medical image analysis. *European radiology experimental, 8(1),* p.26.
Ragab, M., Basheri, M., Albogami, N.N., Subahi, A., Abdulkader, O.A., Alaidaros, H., Mousa, H. and AL-Ghamdi, A.A.M. (2025). Artificial intelligence driven cyberattack detection system using integration of deep belief network with convolution neural network on industrial IoT. *Alexandria Engineering Journal*, *110*, pp.438-450.
Rahmati, M. (2025). Federated learning-driven cybersecurity framework for iot networks with privacy-preserving and real-time threat detection capabilities. *arXiv preprint arXiv:2502.10599*.
Rathnamala, S., Vijayashanthi, T., Prabhananthakumar, M., Panthakkan, A., Atalla, S. and Mansoor, W. (2025). Enhanced Hybrid Deep Learning Approach for Botnet Attacks Detection in IoT Environment. *arXiv preprint arXiv:2502.06138*.
Sharma, S.B. and Bairwa, A.K. (2025). Leveraging AI for Intrusion Detection in IoT Ecosystems: A Comprehensive Study. *IEEE Access*.
Shimaoka, A.M., Ferreira, R.C. and Goldman, A., (2024). The evolution of CRISP-DM for data science: Methods, processes and frameworks. *SBC Reviews on Computer Science*, *4*(1), pp.28-43.
SOW, T.H. and Adda, M., (2024). Enhancing Ids Performance Through a Comparative Analysis of Random Forest, Xgboost, and Deep Neural Networks. *Xgboost, and Deep Neural Networks*.