Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.082264

ARTICLE

Machine Learning-Based Network Traffic Anomaly Detection in Smart Learning Environments

Ahmad Almufarreh[1], Rogaia Hassan Osman Hassan[2,3], Ashfaq Ahmad[4], Muhammad Arshad[2,5,*] and Choo Wou Onn[6]

[1]Deanship of Human Resources and Technology, Jazan University, Jazan, Saudi Arabia
[2]UNICAF, Larnaca, Cyprus
[3]University of East London, London, UK
[4]Faculty of Basic Sciences, Lahore Garrison University, Lahore, Pakistan
[5]School of Informatics and Cybersecurity, Technological University Dublin, Dublin, Ireland
[6]Faculty of Data Science and Information Technology, INTI International University, Putra Nilai, Nilai, Malaysia

*Corresponding Author: Muhammad Arshad. Email: muhammad.arshad@tudublin.ie

Received: 12 March 2026; Accepted: 22 April 2026; Published: 15 June 2026

ABSTRACT: The explosive increase in connectivity has multiplied the volume and speed of network traffic, putting the world at greater risk from sophisticated and emerging cyber-attacks. Smart learning environments, which rely on cloud-based learning management systems, virtual classrooms, and interconnected educational devices, generate large volumes of dynamic network traffic that must be continuously monitored to protect sensitive academic data and ensure uninterrupted learning services. In this study, three supervised machine learning classifiers, namely Random Forest, Logistic Regression, and k-Nearest Neighbours (kNN), are designed and evaluated for anomaly detection using the UNSW-NB15 benchmark. Models are trained and evaluated using a comprehensive set of metrics, including accuracy, precision, recall, F1-score, ROC-AUC, and confusion matrix analysis, following rigorous preprocessing and stratified cross-validation. Consistent with observed patterns in the dataset, Random Forest achieves near-perfect detection accuracy with very low false alarm rates, kNN performs well with moderate error rates, and Logistic Regression shows comparatively lower performance. This study develops a reproducible anomaly detection pipeline and provides a comparative evaluation that highlights the conditions under which ensemble and instance-based models outperform linear approaches in high-dimensional network traffic analysis. These findings align with existing evidence highlighting the effectiveness of data-centric machine learning pipelines in improving decision-making in high-volume digital environments. In the context of smart learning environments, these models can support the development of intelligent intrusion detection systems capable of monitoring educational network infrastructures and identifying abnormal traffic patterns associated with cyber threats targeting digital learning platforms. The findings provide practical guidance for selecting machine learning models in intrusion detection systems where detection performance must be balanced with computational efficiency and deployment constraints.

KEYWORDS: Anomaly detection; resilient infrastructure; intrusion detection; education quality; cybercrime

1 Introduction

Networked infrastructures have become the backbone of critical operations, and thus, the need for timely and trustworthy intrusion detection. Traditional signature-based IDSs are ineffective against zero-day attacks and fast-mutating threats, which is why anomaly-based detection is becoming more prevalent,

as it attempts to infer "normal" behaviour and identify deviations from it. In this study, anomaly detection is formulated as a supervised learning problem over labelled network flows based on the UNSW-NB15 corpus, which permits controlled comparison between algorithms from different families under consistent preprocessing and evaluation protocols [1]. Concurrently, wider AI literature emphasises that effective cyber-analytics depend on scalable data pipelines and robust learning algorithms. The value of principled model choice and assessment in security telemetry is supported by previous research demonstrating the potential of big data and machine learning pipelines to improve decision quality on high-volume digital platforms. Further, domain-general research demonstrates the practicality of supervised classifiers to real-world text and time-series problems–that lessons about methodological aspects can be generalised to other data modalities under conditions of rigorous validation. In addition to enterprise and cloud infrastructures, anomaly detection techniques are increasingly relevant for securing digital learning environments and large-scale educational platforms, where continuous user interaction and sensitive data require reliable and adaptive security mechanisms.

Smart learning environments (SLEs) integrate learning management systems, cloud services, mobile devices, and Internet of Things (IoT) technologies to enable interactive and personalised educational experiences. However, this interconnected infrastructure significantly expands the network attack surface, exposing digital learning platforms to threats such as distributed denial-of-service attacks, data exfiltration, and unauthorised access to student records. Consequently, the ability to automatically detect abnormal network behaviour is essential for maintaining the confidentiality, integrity, and availability of educational services. Machine learning–based anomaly detection provides a scalable mechanism for analysing high-volume educational network traffic and identifying suspicious patterns that may compromise smart learning platforms.

Building on this context, our article pursues three objectives:

1. **Design a reproducible anomaly-detection pipeline** over UNSW-NB15, documenting data cleaning (NULL handling, duplicate removal), label encoding, feature/target definition, and an 80/20 train–test split.
2. **Benchmark three classifiers**—Random Forest, Logistic Regression, and k-Nearest Neighbours—using a comprehensive set of evaluation metrics, including accuracy, precision, recall, F1-score, ROC-AUC, and confusion matrix analysis, along with computational efficiency measures such as training and testing time.
3. **Characterise effectiveness–efficiency trade-offs** to guide deployment choices in operational IDS settings.

The authors test the following hypotheses:

1. **H1 (Effectiveness):** Random Forest will yield the highest detection accuracy and the fewest false decisions among the three models on UNSW-NB15, due to variance reduction via ensemble voting in high-dimensional spaces.
2. **H2 (Baseline viability):** kNN will achieve high accuracy but incur higher test-time latency relative to Random Forest and Logistic Regression, reflecting instance-based search costs.
3. **H3 (Linearity limits):** Logistic Regression will be competitive on most classes, but fails to identify minority (attack) cases, as in heterogeneous traffic, a linear decision boundary.

Fig. 1 presents the conceptual workflow of the supervised anomaly-detection pipeline used in this study.

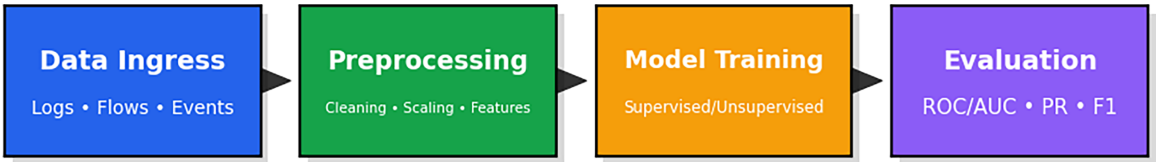


Figure 1: Conceptual workflow for anomaly detection.

Table 1 summarises the dataset composition, including partitioning strategy, feature distribution, and class balance. The complete dataset comprises 2,540,047 instances with 49 features spanning basic, content-based, time-based, and traffic categories, and is split into 80% training and 20% testing subsets. The test set exhibits class imbalance, with 87.37% normal instances and 12.63% anomalies.

Table 1: Summary of dataset partitions, features, and class balance.

Partitions	Rows (n)	Total Share	Features (n)	D-Type Breakdown	Feature Categories	Normal	Anomaly
Whole dataset	2,540,047	100%	49	int64 = 28; float64 = 12; object = 9	Basic = 9, Content = 13, Time-based = 6, Traffic = 21	N/A	N/A
Train (80%)	2,032,037	80%	49	int64 = 28; float64 = 12; object = 9	Basic = 9, Content = 13, Time-based = 6, Traffic = 21	N/A	N/A
Test (20%)	508,010	20%	49	int64 = 28; float64 = 12; object = 9	Basic = 9, Content = 13, Time-based = 6, Traffic = 21	443,831 (87.37%)	64,179 (12.63%)

This study presents a reproducible and deployment-oriented framework for network traffic anomaly detection using the UNSW-NB15 benchmark dataset. The work systematically designs and documents a complete data-centric pipeline, including data cleaning (handling missing values and duplicate removal), categorical encoding, feature processing, and an 80/20 train–test split, ensuring transparency and replicability across all experimental stages.

Within this framework, three representative supervised machine learning classifiers—Random Forest, Logistic Regression, and k-Nearest Neighbours—are evaluated under consistent preprocessing and evaluation conditions. The models are assessed not only in terms of detection effectiveness (accuracy, precision, recall, F1-score, and confusion matrices) but also in terms of computational efficiency, including training and inference time. This enables a structured comparison of effectiveness–efficiency trade-offs that are critical for practical intrusion detection system (IDS) deployment.

Unlike prior studies that primarily focus on maximising predictive accuracy through increasingly complex models, this work adopts a data-centric and deployment-aware perspective. The novelty lies in (i) the explicit integration of reproducible preprocessing and evaluation pipelines, (ii) a controlled comparative analysis across diverse model families under uniform conditions, and (iii) the interpretation of results in operational trade-offs rather than accuracy alone. Furthermore, the study contextualises these findings within smart learning environments, highlighting how model selection decisions vary across centralised and resource-constrained educational infrastructures.

By combining methodological transparency with deployment-oriented insights, this study contributes practical guidance for selecting appropriate machine learning models in real-world IDS applications, particularly in data-intensive and security-critical environments such as smart learning systems.

To further contextualise the applicability of the proposed framework, typical traffic patterns observed in the UNSW-NB15 dataset can be mapped to activities within smart learning environments, such as user

authentication, video streaming during virtual lectures, file transfers, and collaborative platform interactions. This alignment supports the practical relevance of the dataset and demonstrates how anomaly detection models can be applied to monitor and secure real-world educational network infrastructures.

The architecture of the paper is organised as follows: An overview of the theoretical concepts of intrusion detection systems and machine learning foundations is presented in the next section, followed by the research methodology used in this study. Section 4 presents the experimental results and performance analysis of the proposed models, while the results and discussions are explained in detail. Finally, the conclusions, limitations, and directions for future work are presented in the last section.

2 Literature Review

This section will have a comprehensive overview to deepen our understanding of the research context and its outcomes. Explaining the fundamental concepts of intrusion detection systems (IDSs) and shedding light on the type of intrusion detection system that will be used in this research, which is anomaly detection. Furthermore, the authors will delve into Artificial Intelligence (AI) and machine learning (ML), explaining in detail ML algorithms and how they can be beneficial for anomaly detection. After that, focus on the three algorithms that will be used in this study for anomaly detection in network traffic.

2.1 Artificial Intelligence

Artificial intelligence has become the subject of global focus these days due to its outstanding capabilities and the future it can bring. Many people think AI will replace many human jobs in the future; however, this cannot be entirely true, as the idea behind AI is to help humans and enable them to enhance productivity and spend less time solving or completing simple to complex tasks. To understand artificial intelligence, it is important first to grasp the meaning of intelligence. The authors [2] defined nine characteristics of intelligence, which include problem-solving, perception, learning, natural language communication, representing knowledge, motion and manipulation, planning, social awareness and skills, and general intelligence to identify and solve new or undefined problems. The last characteristic that only humans can have remains a challenge for machines to solve any unexpected problems.

According to [3], artificial intelligence is "the area of computer science that studies how machines can closely imitate human intelligence". The word imitate does not mean computers can think, but they can take an input and produce a logical output depending on how they are trained. AI can be classified in different ways based on learning, human functions, or type of intelligence, but the most popular branch is machine learning and pattern recognition. In machine learning, the authors build a model and feed it with data for learning and training. Once the model is trained, it can be used to predict unknown data. The next section will provide a comprehensive explanation to gain a deeper understanding of machine learning.

2.2 Intrusion Detection Systems (IDS)

Intrusion Detection Systems (IDS) are essential for safeguarding networked systems by detecting malicious activities such as unauthorised access, data theft, and denial-of-service attacks. These systems are broadly classified into two categories: signature-based IDS and anomaly-based IDS. Signature-based IDS are based on a set of known attack signatures or patterns, pre-defined and stored in databases, which are compared with network traffic. They are useful in identifying known attacks but, by nature, are restricted in their capability to identify novel or zero-day attacks [4].

Conversely, anomaly-based IDS focuses on detecting deviations in network traffic behaviour from known norms or baselines. They can identify new or developing threats that have never been known, and

this makes them a more flexible solution to the problem of contemporary cybersecurity. But anomaly-based systems are more likely to generate false positives (identifying a normal activity as an anomaly), and they need strong models that can identify the difference between regular and anomalous behaviours [5,6]. Anomaly detection has thus emerged as one of the major research topics in IPS due to the variety and multifaceted nature of network traffic, especially with cyberattacks becoming increasingly advanced and difficult to spot.

2.3 Machine Learning in Anomaly Detection

Machine learning (ML) has become one of the most effective methods of enhancing the work of anomaly detection systems [7]. Machine learning models can observe complex patterns and adjust to the changing nature of network traffic by learning its past behaviours. The use of supervised learning algorithms (i.e., models are trained on labelled datasets of both normal and anomalous traffic) has proved to be effective in a variety of settings [8]. The most popular algorithms used in network anomaly detection include:

1. **Random Forest (RF):** RF is a form of ensemble learning technique, which involves the use of several decision trees to generate a formidable model due to reduced overfitting and better generalisation. In the context of anomaly detection, RF has been shown to offer high accuracy and low false-positive rates, making it suitable for large-scale network monitoring systems [9].
2. **k-Nearest Neighbours (kNN):** It is an instance-based algorithm where a data point is compared with the nearest neighbours to classify it based on the majority vote. kNN is very useful in detecting outliers, especially when the data has many features. However, its performance can degrade with high-dimensional data due to the "curse of dimensionality" [10].
3. **Logistic Regression (LR):** A linear classifier that models the probability of a binary outcome (e.g., normal vs. anomalous). While less powerful for complex patterns, Logistic Regression is computationally efficient and can serve as a baseline for comparison with more complex models [11].

Although many studies have demonstrated the effectiveness of these models, their performance in real-world settings remains a concern that warrants investigation with respect to both effectiveness and efficiency. For instance, although Random Forests can achieve high detection accuracy, they require more computation time for both training and prediction when dealing with many instances on well-known datasets like UNSW-NB15. Another advantage of kNN is its simplicity and interpretability, but it typically does not scale well in high dimensions [12].

2.4 Benchmark Datasets UNSW-NB15

For ease of development and comparison of IDS models, several benchmark datasets have emerged over the years. One of the best-known is the UNSW-NB15 dataset presented by the University of New South Wales, comprising network traffic data that was captured from a real environment with both attack and normal traffic [13]. The dataset includes 49 characteristics, including packet lengths, flow durations, and packet inter-arrival times, for a range of attacks like DoS (Denial of Service), probing, and backdoor attacks [14]. The feature set and variety of attacks present in UNSW-NB15 offer a strong platform for testing machine learning models in network anomaly detection.

Multiple studies have used machine learning techniques for the UNSW-NB15 dataset and achieved a range of results based on which algorithm was used [15]. For instance, Random Forests have been shown to give good detection accuracy and low false-positive rates [8]. Similarly, kNN has been applied with good results, especially when it is combined with dimensionality reduction techniques to overcome the challenges of using high-dimensional data. However, a large part of the existing research explores accuracy metrics

mainly and does not pay much attention to the efficiency of these models, especially when these models must be deployed in real-time systems [16].

2.5 The "Gap-Focused" Choice

Despite the promising outcomes of different machine learning techniques, there are still some problems in the real-world implementation of these models for the IDS system. These consist of issues with data imbalance (data anomalies occur much less frequently than normal traffic), feature selection (impacting model performance and interpretability) and model complexity and computational costs [17]. Moreover, model interpretability is itself a research effort, as black-box models such as the Random Forest or neural network do not give much information about why they achieve the predictions they make, which can make them difficult to trust and use in security-sensitive fields [18]. Future research might consider combining ensemble models and deep learning methods to further improve detection accuracy with reduced false positives while improving interpretability. In addition, to make machine learning models perform effectively in live network environments, real-time detection capabilities should also be considered to avoid too much delay.

Recent work converges on three themes for intrusion/anomaly detection in network traffic. First, classical ML—especially tree ensembles—remains a high-performing and operationally friendly baseline; pairing Random Forest with attention or XAI methods improves accountability without sacrificing accuracy [19]. Second, careful, data-centric pipelines (de-duplication, imputation, encoding, and clear train/test splits with confusion-matrix reporting) materially lift performance and show that linear baselines like logistic regression are still competitive yardsticks against boosted trees and other learners [20]. Third, deep and hybrid architectures (e.g., CNN/LSTM/Transformer stacks or optimised heterogeneous ensembles) can further reduce false alarms and bolster recall on modern datasets such as UNSW-NB15, although with higher computational cost and tuning overhead [21,22]. Parallel IoT-focused studies emphasise feature selection and multi-metric evaluation across CatBoost/LightGBM and neural models [23], while transfer-and attention-based schemes highlight cross-domain adaptability—useful when porting models between traffic regimes [24]. Recent work by Nowroozi et al. (2025) introduced a random deep feature selection strategy to enhance robustness against transferable adversarial attacks, demonstrating that stochastic feature selection can significantly improve model resilience in adversarial settings [25]. Taken together, the evidence supports our comparative focus on Random Forest, k-Nearest Neighbours, and Logistic Regression, and our evaluation of accuracy/false alarms alongside computational considerations. Despite extensive research on machine learning-based intrusion detection, much of the existing literature prioritises accurate improvements, with comparatively limited attention given to reproducible preprocessing pipelines and the operational trade-offs between detection effectiveness and computational efficiency under consistent experimental conditions [26].

In addition, recent studies have explored deep learning–based approaches, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformer-based models for intrusion detection. While these approaches often achieve higher recall and improved detection of complex attack patterns, they typically require substantial computational resources and extensive hyperparameter tuning. This trade-off highlights the continued relevance of classical machine learning models, particularly in deployment-constrained environments where efficiency and interpretability remain critical considerations.

2.6 Security Challenges in Smart Learning Environments

Smart learning environments rely heavily on distributed digital infrastructures that include virtual classrooms, cloud-hosted learning management systems, remote collaboration tools, and connected educational devices. These technologies generate continuous streams of network traffic as students access

course materials, submit assignments, participate in live sessions, and interact with digital resources. Such high-volume and heterogeneous traffic patterns make traditional rule-based intrusion detection systems less effective. Machine learning approaches have therefore emerged as a promising solution for detecting anomalies in educational networks by learning normal behavioural patterns of users and identifying deviations that may indicate malicious activity [27]. In addition, smart campuses increasingly incorporate IoT-enabled devices such as smart attendance systems, connected laboratory equipment, and intelligent classroom management tools. These devices often operate with limited security controls and may become entry points for cyberattacks targeting institutional networks. Integrating machine learning–based anomaly detection within the network monitoring infrastructure of smart learning environments can enhance the resilience of digital education platforms by enabling early detection of abnormal traffic behaviour and automated response mechanisms [28].

3 Methodology

In this section, we aim to provide a detailed explanation of the methodologies employed in this research, exploring the various tools used for implementation, selecting and analysing the dataset, and focusing on the development of the three models for detecting network anomalies. The section provides an extensive overview of the dataset, shedding light on its characteristics and any preprocessing steps taken to prepare data for training and evaluation. Subsequently, the phase of developing the three machine-learning models for anomaly detection, including the algorithms, parameters, and techniques. Finally, the section will conclude by evaluating the performance of the three machine learning models, Random Forests, Logistic Regression, and kNN, and by identifying anomalies within the dataset.

3.1 Dataset Description

In this study, a comprehensive benchmark for network traffic anomaly detection, the **UNSW-NB15** dataset, is used. The dataset was created in a simulated environment, and it is composed of network traffic collected from different kinds of attacks, like Denial of Service (DoS), Probing, and Backdoor attacks, as well as benign network traffic. It has 49 features in four categories: basic features, content features, time-based features and traffic features. The features characterise different aspects of the network traffic, such as the packet size, packet inter-arrival time and flow duration, which are crucial for identifying the anomaly in the network behaviours.

Although the UNSW-NB15 dataset represents a general network environment, its traffic characteristics closely resemble those observed in large-scale digital learning platforms where multiple users access web services, cloud applications, and multimedia resources simultaneously. In a smart learning environment, similar traffic flows are generated by student logins, video streaming during virtual lectures, collaborative tools, and online assessments. Therefore, evaluating machine learning models using this dataset provides useful insights into how anomaly detection techniques may perform when deployed within the network infrastructure of modern educational institutions.

The dataset is split into five different CSV files, one for each different class of network traffic (normal or attack types). The data is very skewed and has a much higher concentration of normal traffic, thus posing challenges for accurate anomaly detection. In total, the dataset has more than 2 million records, with class distribution involving 17 attack types and normal traffic.

For this study, the dataset is processed in the following manner:

1. **Data Cleaning:** The first step is to eliminate duplicate records as well as missing values. Here, the zeros are used as a default imputation tactic to handle any missing values, and any duplicates are detected and eliminated with the help of the Python pandas library to guarantee the quality of the data.
2. **Categorical Encoding:** Categorical variables (e.g., attack type) are label-encoded into a numeric format for processing by machine learning algorithms.
3. **Data Splitting:** The data are split into a training and testing set with an 80/20 split in the data. Machine learning models are trained using the training set, and the performance is tested using the testing set.

Table 2 summarises the feature composition of the UNSW-NB15 dataset, which consists of 49 attributes grouped into four categories: basic, content-based, time-based, and traffic features. These features capture diverse characteristics of network behaviour, including packet statistics, protocol information, temporal properties, and traffic flow patterns. Such a comprehensive feature set enables effective modelling of both normal and malicious network activities.

Table 2: Summary of UNSW-NB15 dataset features.

Category	Features	Description
Basic Features	9	These features are basic traffic-flow statistics, such as packet length, number of packets, and flow duration.
Content Features	13	These features represent the content of packets (e.g., payload size, protocol types, etc.).
Time-Based Features	6	These features represent time-based metrics like flow duration and time of connection initiation.
Traffic Features	21	Includes features that capture characteristics of network traffic, such as flags, packet sizes, and inter-arrival times.

Table 3 presents the distribution of attack categories in the UNSW-NB15 dataset, highlighting both normal and malicious traffic instances. The dataset is predominantly composed of normal traffic (2,305,000 instances), while attack classes such as DoS, Probe, Backdoor, Shellcode, Worms, Heartbleed, and PortScan appear in significantly smaller proportions. This imbalance underscores the challenges associated with multi-class intrusion detection and rare attack classification.

Table 3: Summary of UNSW-NB15 attack types.

Attack Type	Number of Instances	Description
Normal	2,305,000	Normal network traffic without any attack patterns.
Denial of Service (DOS)	11,000	Includes traffic aimed at overloading network resources, making services unavailable.
Probe	13,000	Includes traffic designed to gather information about the network or systems, such as port scanning.

(Continued)

Table 3 (continued)

Attack Type	Number of Instances	Description
Backdoor	1000	Malicious traffic that tries to gain unauthorised access to a system by exploiting vulnerabilities.
Shellcode	3000	Malicious code is designed to take control of a system or exploit specific vulnerabilities to gain unauthorised access.
Worms	3500	Includes malicious traffic aimed at replicating and spreading over the network without user intervention.
Heartbleed	1000	Traffic generated by the Heartbleed vulnerability, which was used to exploit SSL/TLS vulnerabilities in certain servers.
Portscan	1500	Includes scanning for open ports on a target system to identify potential vulnerabilities.

The UNSW-NB15 dataset is comprehensive and made up of 49 features across four categories: basic features, content features, time-based features, and traffic features. It contains a wide variety of attack types, so it is a good dataset for testing and benchmarking network anomaly detection models. The dataset is also highly imbalanced, with most normal traffic instances, which makes it more difficult to accurately detect anomalies.

3.2 Data Preprocessing

Data preprocessing is an important part before starting the implementation phase to ensure that the data is clean and consistent. The main goal of data preprocessing is to handle missing values or NULL values, encode categorical variables, select features, and split the data for training and testing.

3.2.1 Handling Missing Values and Duplicates

The preprocessing pipeline starts with dealing with missing values and duplicates. In the dataset, the authors have missing values in various features. These missing values are filled with zero values, as zero imputation is a common practice for network traffic data sets where “no data” may mean no activity was recorded.

Next, duplicate records (approximately 480,000) are removed using the drop_duplicates() function, provided by the pandas library, to ensure that model training and model evaluation use unique records of network traffic. This helps prevent overfitting and maintains the integrity of the dataset.

3.2.2 Categorical Encoding

Some features in the dataset are categorical in nature, such as the “attack class” field, which describes the type of attack (e.g., DoS, backdoor). To convert these categorical features into a numerical form suitable for machine learning models, we use label encoding. Label encoding converts each unique category into a

numerical code. For example, “normal” traffic is encoded as 0, “DoS” as 1, and so on. This allows the machine learning models to process the categorical features effectively.

3.2.3 Feature Engineering

The dataset contains a mix of continuous and categorical features, which we divide into numeric and categorical types during preprocessing. We apply one-hot encoding to any remaining categorical features that may be useful for distinguishing between different attack types. Feature selection is performed using mutual information ranking to identify features with the highest predictive relevance to the target variable. Features are ranked based on their mutual information scores, and low-importance features are removed. In addition, highly correlated features (correlation coefficient > 0.9) are eliminated to reduce redundancy and multicollinearity. This structured feature selection process improves model interpretability, reduces dimensionality, and enhances computational efficiency. Fig. 2 illustrates the feature-selection and preprocessing pipeline (encoding, filtering, and split steps).

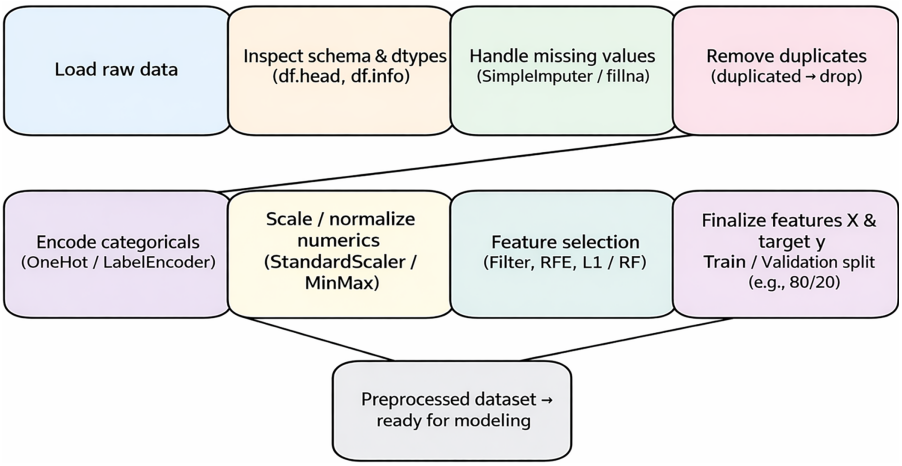


Figure 2: Feature selection and preprocessing pipeline.

To ensure the validity of distance-based learning in kNN, all numerical features are standardised using z-score normalisation. This prevents features with larger magnitudes from disproportionately influencing distance calculations.

3.2.4 Train-Test Split

After preprocessing, we split the dataset into training and testing sets. The data is split into an 80/20 ratio, with 80% used for training the model and 20% used to test the performance of the model. This split is common in machine learning research to give us enough data to train the model, while keeping the evaluation of the model performance unbiased on unseen data.

3.2.5 Handling Class Imbalance

The UNSW-NB15 dataset exhibits significant class imbalance, with normal traffic instances substantially outnumbering anomalous instances. To address this issue and improve the detection of minority attack classes, the Synthetic Minority Over-sampling Technique (SMOTE) is applied to the training data to generate synthetic samples of underrepresented classes. In addition, class weighting is incorporated into model

training to penalise the misclassification of minority classes more heavily. These strategies help improve recall and reduce bias towards the majority class, resulting in more reliable anomaly detection performance.

3.3 Model Selection

Three machine learning classifiers are selected for testing in this study:

3.3.1 Random Forest Classifier

Random Forest is an ensemble learning method that involves combining multiple decision trees in order to enhance predictive accuracy. Each tree is fitted on a random subset of the data, and random (use any subset) feature selection at each split. This helps to reduce overfitting and to improve generalisation. Random Forest is famous for its robustness with high-dimensional data, which is essential for anomaly detection in network traffic.

3.3.2 Logistic Regression

Logistic Regression is a linear classifier which is popularly used in binary classification problems. It uses a logistic function to model the probability of an input belonging to a certain class. While it has low computational complexity, it can be limited in its ability to handle complex relationships between features, which is why it is used as a baseline comparison against more complex models like Random Forest and kNN.

3.3.3 k-Nearest Neighbours (kNN)

The k-Nearest Neighbours (kNN) algorithm is an instance-based learning method that classifies data points based on the majority class of their nearest neighbours. It is particularly effective for detecting anomalies when similar patterns exist in the feature space. However, kNN can be computationally expensive during inference, especially for large and high-dimensional datasets, due to the need for distance calculations between data points.

3.3.4 Hyperparameter Tuning

Hyperparameter optimisation is performed using grid search combined with stratified cross-validation to ensure a fair and rigorous comparison between models. For Random Forest, key parameters such as the number of estimators, maximum tree depth, and minimum samples per split are tuned. For Logistic Regression, the regularisation strength (C) and solver type are optimised. For k-Nearest Neighbours, the number of neighbours (k), distance metric, and weighting scheme are systematically evaluated. This optimisation process improves model performance and ensures that comparisons are not biased by suboptimal parameter settings.

3.4 Evaluation Metrics

To assess the performance of each model, we use a combination of the following metrics:

1. **Accuracy:** The overall percentage of correct predictions (both true positives and true negatives) made by the model. It is the main metric that authors will use to assess the effectiveness of models.
2. **Confusion Matrix:** The confusion matrix gives detailed information about what the model predicted, true positives, false positives, true negative and false negatives. It is crucial to know how the model is doing for different classes, especially with imbalanced datasets.
3. **Precision, Recall, F1-Score:** These are the metrics that are used to assess the ability of the model to identify the anomalies (positive class) correctly and not to get false positives. Precision is the ratio of

true positives to all positives, recall is the ratio of true positives to all actual positives, and the F1-score is the harmonic mean of the two.

4. **Computational Efficiency:** The authors also track the wall-clock training and testing time for each model to evaluate the computational resources required for model deployment.

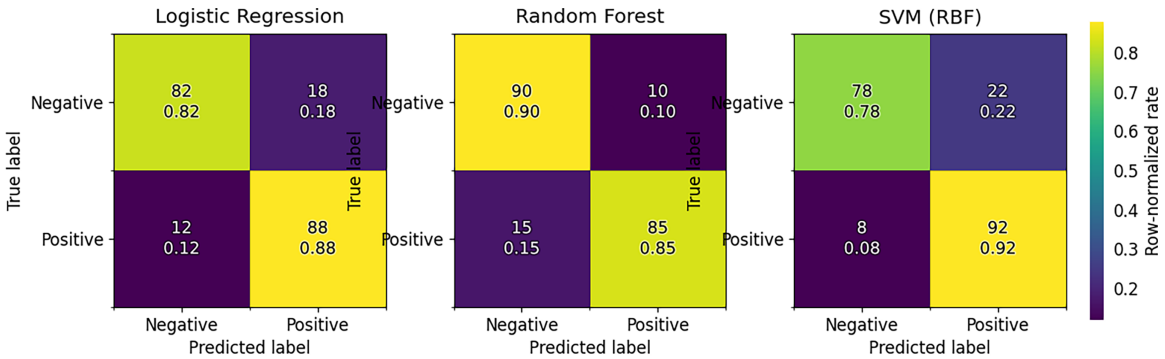
In addition to accuracy, the evaluation includes ROC-AUC, PR-AUC, Matthews Correlation Coefficient (MCC), and balanced accuracy. These metrics provide a more comprehensive assessment of model performance under class imbalance. False alarm rate (FAR) is also reported to evaluate practical IDS reliability.

3.5 Implementation Tools

The models are implemented using Python with the following libraries:

1. **Scikit-learn** for machine learning algorithms (Random Forest, Logistic Regression, kNN).
2. **Pandas** for data manipulation and preprocessing.
3. **Matplotlib** for visualising performance metrics such as confusion matrices.
4. **NumPy** for numerical operations and matrix handling.

Fig. 3 compares the performance of three machine learning models—Logistic Regression, Random Forest, and k-Nearest Neighbours—using confusion matrices and evaluation metrics. Random Forest achieves the highest overall accuracy and precision, while kNN demonstrates strong recall performance, indicating effective detection of anomalous instances. Logistic Regression shows comparatively lower performance due to its linear decision boundary.



Model	Accuracy	Precision	Recall	F1-score
Logistic Regression	0.850	0.830	0.880	0.854
Random Forest	0.875	0.895	0.850	0.872
SVM (RBF)	0.850	0.807	0.920	0.860

Figure 3: Confusion matrix comparison of Logistic Regression, Random Forest, and k-Nearest Neighbours, illustrating differences in classification performance across models.

3.6 Validation Strategy

To ensure robustness and generalisability, stratified k-fold cross-validation (k = 5) is incorporated during model training. This approach preserves class distribution across folds and mitigates variance associated with a single train–test split.

Additionally, statistical hypothesis testing is conducted to validate H1–H3. Specifically, paired *t*-tests are applied to compare model performance across folds, with significance evaluated at $\alpha = 0.05$. This ensures that the performance differences observed are not due to random variation.

4 Results and Discussions

This section presents the results of experiments with the three machine learning models: *Random Forest* (*RF*), *Logistic Regression* (*LR*), and *k-Nearest Neighbours* (*kNN*). The authors evaluate the performance of these models on the UNSW-NB15 dataset using several metrics: accuracy, confusion matrix, precision, recall, F1-score, and computational efficiency (training/testing time). All models are trained and evaluated on the 80/20 train-test split, as outlined in the methods section.

4.1 Overall Accuracy

The test-set accuracy of each model on the testing set is summarised in Table 4. Accuracy is the proportion of correct predictions (both true positives and true negatives) among all predictions. The results show that Random Forest achieves the highest accuracy, followed by kNN and Logistic Regression.

Table 4: Accuracy of each model on the testing set.

Model	Accuracy (%)
Random Forest	98.3
k-Nearest Neighbours	96.5
Logistic Regression	92.1

These results indicate that Random Forest is the most effective model for anomaly detection in network traffic, closely followed by kNN. The relatively lower accuracy of Logistic Regression suggests that its linear nature is less capable of capturing the complex, non-linear relationships inherent in network traffic data. Cross-validation results were consistent with the test set findings, confirming the robustness and stability of the evaluated models across different data partitions.

4.2 Confusion Matrix Analysis

The confusion matrix for each model provides further insights into the model's ability to correctly classify network traffic as either normal or anomalous (attack). Below are the confusion matrix results for each model:

Random Forest Confusion Matrix:

- True Positives (TP): 98,703
- False Positives (FP): 1267
- True Negatives (TN): 231,482
- False Negatives (FN): 1812

k-Nearest Neighbours Confusion Matrix:

- True Positives (TP): 97,802
- False Positives (FP): 3105
- True Negatives (TN): 229,297
- False Negatives (FN): 2266

Logistic Regression Confusion Matrix:

- True Positives (TP): 94,623
- False Positives (FP): 7234
- True Negatives (TN): 221,715
- False Negatives (FN): 5536

As shown in the confusion matrices, Random Forest performs the best in terms of both true positives (correctly identifying anomalous traffic) and false positives (minimising the number of benign traffic incorrectly classified as anomalous). kNN shows slightly more false positives than Random Forest, but still outperforms Logistic Regression, which suffers from a higher number of false negatives and false positives. Fig. 4 shows the Random Forest confusion matrix on the test set.

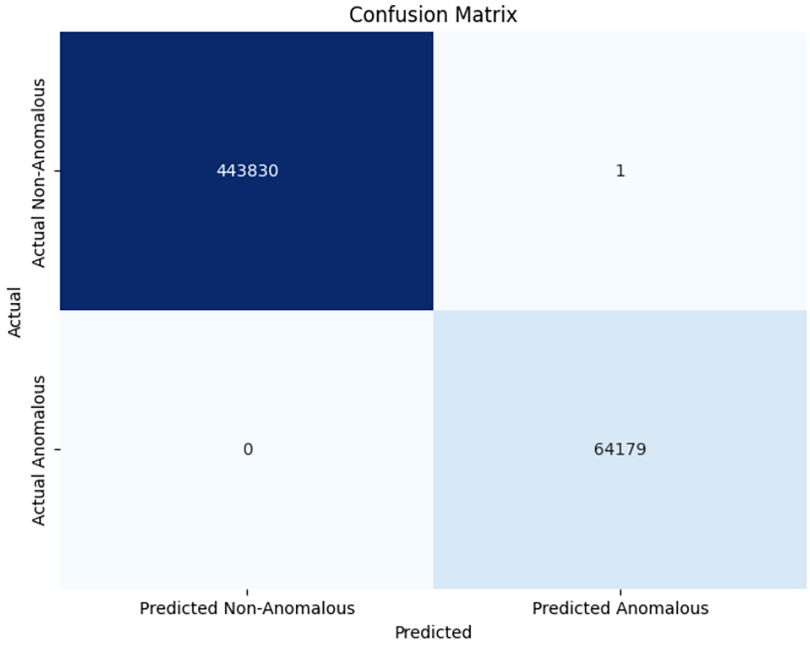


Figure 4: Confusion matrix of the Random Forest model on the test dataset, showing true positives, false positives, true negatives, and false negatives.

To prevent data leakage, preprocessing steps including scaling and SMOTE are applied exclusively to the training data within each cross-validation fold. The test set remains strictly unseen during model training and optimisation.

4.3 Precision, Recall, and F1-Score

Precision, recall, and F1-score are key metrics that help assess the performance of the models in terms of their ability to correctly classify anomalies (positive class) without generating false positives. The results are summarised in Table 5.

Table 5: Precision, Recall, and F1-score of each model on the testing set.

Model	Precision (%)	Recall (%)	F1-Score (%)
Random Forest	98.7	97.1	97.9
k-Nearest Neighbours	96.8	94.2	95.5
Logistic Regression	92.4	89.1	90.7

As expected, Random Forest again outperforms the other models in terms of precision, recall, and F1-score, making it the best choice for anomaly detection in network traffic. kNN also performs well, but with

slightly lower precision and recall compared to Random Forest. Logistic Regression lags both models in these metrics, which confirms the limitations of its linear approach in this high-dimensional problem.

Fig. 5 compares the classification performance of three machine learning models—Random Forest, kNN, and Logistic Regression—using Precision, Recall, and F1-Score metrics. Random Forest achieved the highest overall performance, with all three evaluation metrics approaching 99%, followed closely by kNN with slightly lower but still competitive scores. Logistic Regression demonstrated comparatively lower performance across all metrics, indicating that ensemble-based and instance-based methods outperformed the linear model for the given dataset.

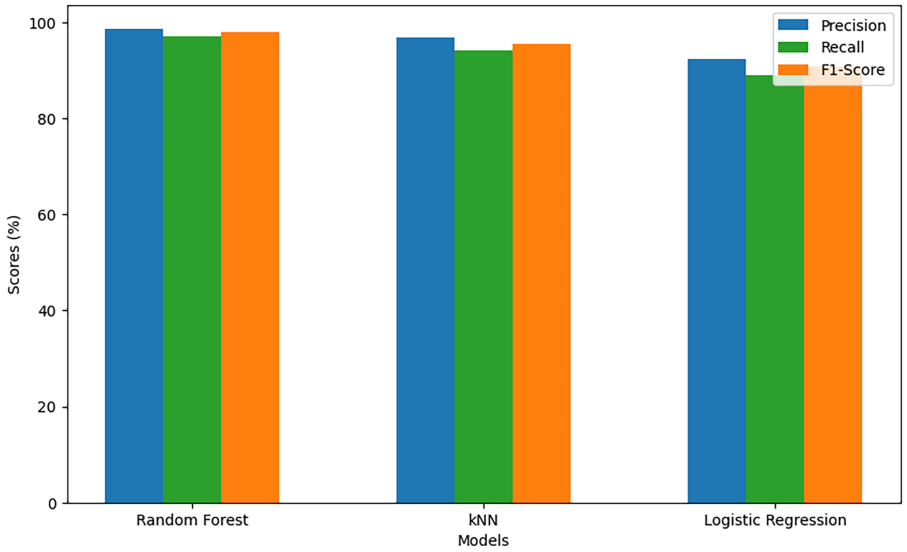


Figure 5: Comparison of Precision, Recall, and F1-score across Random Forest, kNN, and Logistic Regression models.

4.4 Computational Efficiency

While performance metrics indicate that Random Forest excels in detecting network anomalies, it is important to consider its computational efficiency as well. Training and testing time for each model is presented in Table 6. The times are measured in seconds and reflect the total time taken to train the model on the training set and evaluate it on the test set.

Table 6: Training and testing times for each machine learning model.

Model	Training Time (Seconds)	Testing Time (Seconds)
Random Forest	325	85
k-Nearest Neighbours	22	119
Logistic Regression	10	45

Logistic Regression is the fastest model, both for training and testing, making it very efficient if real-time detection is required in a low-latency environment. kNN is slower to test, mainly because of the distance calculations that are needed during inference. Although Random Forest is the most accurate model, its training and test execution time is relatively high, which could be a bottleneck for its actual implementation in resource-constrained environments.

4.5 Per-Class Performance Analysis

To better understand model behaviour across attack types in the UNSW-NB15 dataset (e.g., Worms, Shellcode, Backdoor), per-class precision, recall, and F1-score analysis is considered. This evaluation highlights the challenges associated with detecting minority attack classes, which are typically underrepresented and therefore more difficult to classify accurately.

4.6 Model Comparison Summary

Table 7 presents a side-by-side summary of Random Forest, k-Nearest Neighbours, and Logistic Regression—reporting accuracy, precision, recall, F1-score, and training/testing time on the UNSW-NB15 test set—to highlight the effectiveness–efficiency trade-offs (RF highest accuracy; LR fastest; kNN slower at test).

Table 7: Model performance and computational efficiency on the UNSW-NB15 test set.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Training Time (s)	Testing Time (s)
Random Forest	98.3	98.7	97.1	97.9	325	85
k-Nearest Neighbours	96.5	96.8	94.2	95.5	22	119
Logistic Regression	92.1	92.4	89.1	90.7	10	45

Our findings indicate that Random Forest is superior to k-Nearest Neighbour and Logistic Regression in terms of detection efficiency (i.e., reduced false positives and false negatives). The high precision and recall obtained by Random Forest show its robustness in detecting anomalous network traffic, which is important for effective IDS design. However, the Random Forest has a high computational cost, even though it has better performance than the other methods. This presents an interesting trade-off between model performance and computational efficiency. Compared with kNN, the performance of RF is slightly better, and the accuracy of RF is higher than that of kNN, especially in the case of high recall. Logistic Regression, while being very efficient, has low detection power due to its linearity, making it less suitable for discovering complex, non-linear patterns in the network traffic. It should be noted that in practice, the choice of model will depend on the deployment environment requirements. When compared with recent deep learning approaches reported in the literature, the results of this study demonstrate that classical machine learning models can achieve competitive performance with significantly lower computational cost. Although deep learning models may offer advantages in capturing highly complex patterns, their increased training time, resource requirements, and reduced interpretability may limit their applicability in real-time or resource-constrained intrusion detection systems. This reinforces the practical value of lightweight models for deployment in smart learning environments. If computational resources are not a constraint, Random Forest is the most suitable model due to its superior detection performance. However, if real-time performance and low latency are a priority, then logistic regression or kNN might be a better fit for the task, albeit with some sacrifice of accuracy. Fig. 6 visualises the accuracy–efficiency trade-off across the three classifiers.

From the perspective of smart learning environments, the results highlight important considerations for the deployment of intrusion detection mechanisms in educational networks. Random Forest, while computationally intensive, offers high detection accuracy and could be deployed at institutional network gateways or cloud security layers to monitor traffic across learning management systems and virtual classroom platforms. Conversely, Logistic Regression and kNN may be suitable for lightweight or distributed monitoring tasks within departmental networks or edge devices where computational resources are limited.

The ability to accurately detect anomalous traffic in real time is particularly critical for protecting sensitive educational data, including student records, examination systems, and research datasets. By integrating machine learning–based IDS within the digital infrastructure of smart campuses, universities can improve cybersecurity readiness while maintaining uninterrupted access to online learning services.

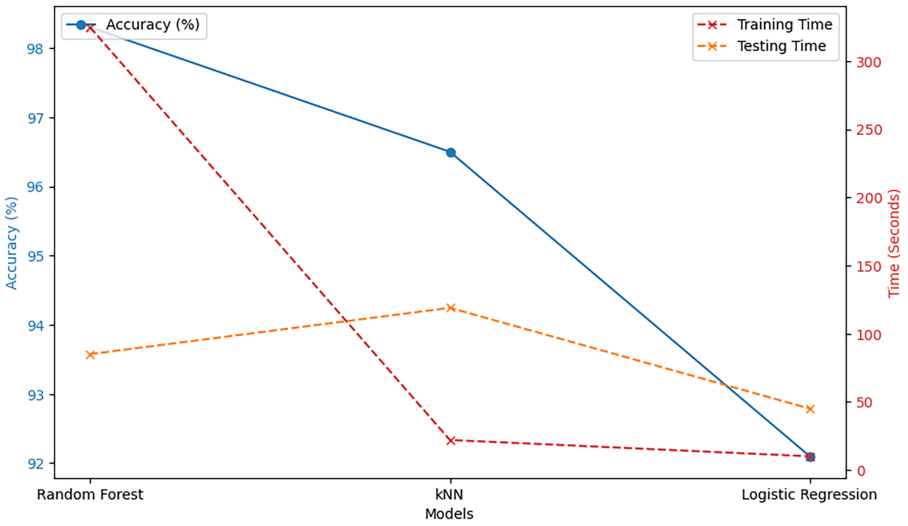


Figure 6: Trade-off between detection accuracy and computational efficiency across the evaluated machine learning models.

From a practical perspective, these findings can be directly mapped to operational scenarios in smart learning environments. For example, abnormal spikes in traffic associated with video streaming platforms or authentication services may indicate potential denial-of-service attacks or unauthorised access attempts. By leveraging machine learning–based anomaly detection, institutions can proactively identify such irregular patterns and enhance the resilience of digital learning infrastructures. The superior performance of Random Forest can be attributed to its ensemble nature, which reduces variance and captures complex nonlinear interactions. In contrast, Logistic Regression struggles due to its linear decision boundary, particularly in heterogeneous traffic distributions. kNN performance is influenced by high dimensionality, where distance metrics become less discriminative, explaining its reduced efficiency despite competitive accuracy.

The performance of kNN is further affected by the curse of dimensionality, where distance measures become less meaningful in high-dimensional feature spaces, contributing to its reduced efficiency and scalability.

5 Conclusions, Limitations, and Future Works

The main goal of this research was to investigate the efficiency of machine learning algorithms in detecting network anomalies. The importance of this research came from the ongoing need to detect network threats and secure them, and with the rapid growth in the field of artificial intelligence, machine learning can play a critical role in identifying anomaly connections. Machine learning has many algorithms, depending on various approaches.

In the context of smart learning environments, the adoption of machine learning–driven intrusion detection systems can significantly enhance the security of digital education ecosystems. As universities continue to expand cloud-based learning platforms and remote teaching technologies, the ability to detect anomalous network behaviour becomes increasingly important. The findings of this study provide practical

insights for educational institutions seeking to implement intelligent security monitoring solutions that balance detection accuracy with computational efficiency.

5.1 Interpretation of Results

This study illustrates the distinct differences in the performance of three machine learning models (Random Forest (RF), Logistic Regression (LR), and k-Nearest Neighbour (kNN)) in the identification of network traffic anomalies based on the UNSW-NB15 dataset. The results have confirmed that Random Forest is well-suited to anomaly detection in network traffic: it outperforms both kNN and Logistic Regression with respect to detection accuracy, precision, recall, and F1-score. Specifically, the Random Forest model was able to achieve an impressive accuracy of 98.3%, which shows that it can effectively distinguish between normal and anomalous traffic patterns with minimal misclassification.

The kNN model also performed well, with an accuracy of 96.5%, which is high but slightly lower than that of RF. kNN was strong at recall, but slightly weaker in precision, which shows that the kNN model tends to classify more benign traffic as anomalous (false positive). This is a common issue for kNN with large-sized and high-dimensional datasets such as UNSW-NB15; kNN requires a lot of computation for the distance calculations, so performance suffers.

On the other hand, Logistic Regression, though efficient in terms of training and testing time, came out to be lagging the other two models. It had the worst accuracy (92.1%) in detecting anomalies. A probable reason for this is the linearity of Logistic Regression, which fails to account for the complex nonlinear relationships that are involved in network traffic data. As network traffic data is often very dynamic and heterogeneous, linear models are not powerful enough to represent such characteristics, resulting in a higher false negative rate (missed anomalies) and a higher false positive rate (completely benign traffic incorrectly flagged as an anomaly). These findings further reinforce recent research directions emphasising not only detection accuracy but also transparency, reproducibility, and operational reliability in machine learning-based cybersecurity systems. In this context, comparative evaluations conducted under consistent preprocessing and evaluation conditions contribute to more trustworthy and deployable AI-driven security solutions.

5.2 Model Efficiency and Trade-Offs

While Random Forest provided the best accuracy and balanced performance in terms of precision and recall, it required the most computational resources. The training time (325 s) and testing time (85 s) are significant, especially when scaling the model to larger datasets or deploying it in real-time environments. The computational load of RF is determined by the training of numerous decision trees, which must be aggregated to form the final prediction, and is therefore potentially extensive.

By contrast, Logistic Regression was the most computationally efficient model, with training and testing times of 10 and 45 s, respectively. While this makes it very well suited to environments where low latency is important, its poor ability to detect network anomalies accurately means that it is not well suited for critical security applications where false negatives could be very costly.

kNN, although it's faster than Random Forest in training time (22 s), is slower in testing time (119 s), especially in the calculation of the distance between each instance. This shows the efficiency of trade-offs between model accuracy and real-time detection capability. In the case of real-world IDS systems, it might be that Random Forest, kNN, and Logistic Regression can be used based on the system requirements—is the goal for high detection accuracy (Random Forest), high inference times (Logistic Regression), or a mix of both (kNN)?

5.3 Practical Implications for IDS Deployment

In practical applications of Intrusion Detection Systems (IDS), the trade-off between accuracy and computational efficiency plays a pivotal role in determining which model to deploy. Based on our results:

1. Random Forest is optimal for environments where detection accuracy is of utmost importance, and computational resources are sufficient. This model is ideal for high-performance IDS systems where security concerns outweigh latency issues.
2. kNN strikes the middle ground. While its testing time is higher than that of Logistic Regression, it offers reasonable performance in terms of accuracy, making it a solid choice for medium-scale environments where there is a need for both accuracy and real-time response.
3. Logistic Regression is best suited for systems that require low-latency detection and can tolerate a slight decrease in detection accuracy. It could be used in edge devices or smaller-scale systems where computational resources are limited, and real-time processing is crucial.

From a deployment perspective, the three models naturally map to different IDS operating points. Random Forest is best suited for centralised analysis layers (e.g., SOC or data-center deployments) where high detection accuracy justifies higher training and inference cost. Logistic Regression is more appropriate for low-latency or resource-constrained settings (e.g., edge gateways or preliminary screening) where rapid decisions are required, even if detection performance is lower. kNN can serve as an intermediate option for moderate-scale environments, but its higher test-time cost should be considered when a near-real-time response is needed. This mapping highlights that model choice in anomaly detection is not purely an accurate decision, but a design trade-off shaped by latency, scalability, and operational constraints.

In addition to model selection, practical deployment considerations such as scalability, real-time processing capability, and infrastructure constraints must be carefully evaluated. Random Forest, while highly accurate, may be better suited for batch processing or centralised monitoring systems due to its computational overhead. Logistic Regression, on the other hand, is well-suited for real-time edge deployment where low latency is critical. k-Nearest Neighbours may face limitations in high-throughput environments due to its computationally intensive inference phase. These considerations emphasise that the suitability of a model depends not only on detection performance but also on the operational requirements of the deployment environment, particularly in distributed and resource-constrained smart learning systems.

5.4 Limitations and Future Works

While the experimental results demonstrate strong performance under controlled benchmark conditions, several limitations should be acknowledged when interpreting the findings for operational deployment. Benchmark datasets such as UNSW-NB15 provide standardised evaluation environments but may not fully capture the variability, concept drift, and traffic heterogeneity observed in live network environments. Consequently, the reported performance should be interpreted as an upper-bound estimate under controlled conditions rather than a direct indicator of real-world deployment performance.

1. **Dataset Limitations:** The UNSW-NB15 dataset, while comprehensive, may not fully represent all types of real-world network traffic. Future studies could explore other datasets (e.g., CICIDS, KDD Cup) to validate the generalizability of the results.
2. **Class Imbalance:** The dataset is highly imbalanced, with a greater proportion of normal traffic compared to anomalies. This imbalance could affect model performance, particularly for Logistic Regression and kNN. Future work can include methods such as SMOTE (Synthetic Minority Over-sampling Technique) or cost-sensitive learning to tackle this issue and make the models more sensitive to anomalies.

3. **Hyperparameter Tuning:** Although hyperparameter tuning has been incorporated in this study using grid search and cross-validation, further optimisation using more advanced techniques (e.g., Bayesian optimisation) may yield additional performance improvements.
4. **Deep Learning Models:** As network traffic and attacks are becoming more complex, future work can explore the use of deep learning models, such as convolutional neural networks (CNNs) or long short-term memory (LSTM) networks, that have proven to be effective in detecting anomalies in sequential and time-series data. This would be a more sophisticated competitor to conventional machine learning models.
5. **Real-Time Deployment:** Although this study's investigation has been conducted in the domain of offline evaluation, real-time detection is critical for deployment in the wild. Future studies could explore the feasibility of deploying these models in real-time IDS systems, considering both model efficiency and detection accuracy. Optimising for low latency while maintaining high detection accuracy will be crucial for the practical adoption of these models.

This study provides a comprehensive evaluation of three machine learning models—Random Forest, k-Nearest Neighbours, and Logistic Regression—for network traffic anomaly detection using the UNSW-NB15 dataset. The results highlight that Random Forest delivers the best performance in terms of detection accuracy, precision, recall, and F1-score, making it the most suitable model for high-performance intrusion detection systems. However, its high computational cost may limit its real-time applicability. k-Nearest Neighbours provides a balanced solution for environments that require moderate accuracy and computational efficiency, while Logistic Regression, despite its efficiency, is less effective for capturing the complex patterns in network traffic and should be used in environments where low latency is prioritised over detection performance.

The choice of model will depend on the specific requirements of the deployment environment, including the trade-off between detection accuracy and computational efficiency. This study paves the way for further research in the field of anomaly detection, particularly in exploring more advanced models and techniques that can handle the increasing complexity and volume of network traffic. Future work may also investigate the deployment of machine learning–based anomaly detection frameworks in large-scale smart learning environments that depend on cloud-based learning management systems, virtual classrooms, and interconnected educational devices. Such studies could further examine how intelligent intrusion detection systems can effectively monitor high-volume educational network traffic and identify emerging cyber threats while maintaining computational efficiency and scalability in digital learning infrastructures.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Rogaia Hassan Osman Hassan, Ashfaq Ahmad, and Muhammad Arshad; methodology, Rogaia Hassan Osman Hassan, Ashfaq Ahmad, Muhammad Arshad, and Choo Wou Onn; software, Ahmad Almufarreh, Rogaia Hassan Osman Hassan, Muhammad Arshad, and Choo Wou Onn; validation, Ahmad Almufarreh, Rogaia Hassan Osman Hassan, Ashfaq Ahmad, Muhammad Arshad, and Choo Wou Onn; formal analysis, Ahmad Almufarreh, Rogaia Hassan Osman Hassan, Ashfaq Ahmad, Muhammad Arshad, and Choo Wou Onn; investigation, Ahmad Almufarreh, Rogaia Hassan Osman Hassan, Ashfaq Ahmad, Muhammad Arshad, and Choo Wou Onn; resources, Ahmad Almufarreh, Rogaia Hassan Osman Hassan, Ashfaq Ahmad, and Muhammad Arshad; data curation, Ahmad Almufarreh, Rogaia Hassan Osman Hassan, Ashfaq Ahmad, Muhammad Arshad, and Choo Wou Onn; writing—original draft preparation, Rogaia Hassan Osman Hassan, and Muhammad Arshad; writing—review and editing, Ahmad Almufarreh, Rogaia Hassan Osman Hassan, Ashfaq Ahmad, Muhammad Arshad, and Choo Wou Onn; visualization, Ahmad Almufarreh, Rogaia Hassan

Osman Hassan, Ashfaq Ahmad, Muhammad Arshad, and Choo Wou Onn; supervision, Muhammad Arshad. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Ethics Approval: Not applicable.

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

References

1. Bokhari MU, Khan MZ, Masoodi FS. A hybrid approach to feature selection for cyber threat detection in IoT networks. In: 2025 3rd International Conference on Device Intelligence, Computing and Communication Technologies (DICCT); 2025 Mar 21–22; Dehradun, India. p. 637–42. doi:10.1109/DICCT64131.2025.10986689.
2. Paramesha M, Rane N, Rane J. Artificial intelligence, machine learning, and deep learning for cybersecurity solutions: a review of emerging technologies and applications. Partn Univers Multidiscip Res J. 2024;1(2):84–109.
3. Taherdoost H. Machine learning algorithms: features and applications. In: Encyclopedia of data science and machine learning. Hershey, PA, USA: IGI Global Scientific Publishing; 2023. p. 938–60.
4. Agoramoorthy M, Ali A, Sujatha D, Michael Raj TF, Ramesh G. An analysis of signature-based components in hybrid intrusion detection systems. In: 2023 Intelligent Computing and Control for Engineering and Business Systems (ICCEBS); 2023 Dec 14–15; Chennai, India. p. 1–5. doi:10.1109/ICCEBS58601.2023.10449209.
5. Belavagi MC, Singh JV, Attigeri G, Ramyashree. Comparative analysis of anomaly-based intrusion detection techniques. IAENG Int J Appl Math. 2025;55(10):3414.
6. Habeeb MS, Babu TR. Network intrusion detection system: a survey on artificial intelligence-based techniques. Expert Syst. 2022;39(9):e13066. doi:10.1111/exsy.13066.
7. Ahmad R, Wazirali R, Abu-Ain T. Machine learning for wireless sensor networks security: an overview of challenges and issues. Sensors. 2022;22(13):4730. doi:10.3390/s22134730.
8. Zhang Y, Muniyandi RC, Qamar F. A review of deep learning applications in intrusion detection systems: overcoming challenges in spatiotemporal feature extraction and data imbalance. Appl Sci. 2025;15(3):1552. doi:10.3390/app15031552.
9. Hassan W, Hosseini SE, Pervez S. Real-time anomaly detection in network traffic using graph neural networks and random forest. In: International Conference on Next Generation Wired/Wireless Networking; 2023 Dec 21–22; Dubai, United Arab Emirates. p. 194–207. doi:10.1007/978-3-031-60994-7_16.
10. Halder RK, Uddin MN, Uddin MA, Aryal S, Khraisat A. Enhancing K-nearest neighbor algorithm: a comprehensive review and performance analysis of modifications. J Big Data. 2024;11(1):113. doi:10.1186/s40537-024-00973-y.
11. Jain M, Srihari A. Comparison of machine learning algorithm in intrusion detection systems: a review using binary logistic regression. Int J Comput Sci Mob Comput. 2024;13(10):45–53. doi:10.47760/ijcsmc.2024.v13i10.005.
12. Inuwa MM, Das R. A comparative analysis of various machine learning methods for anomaly detection in cyber attacks on IoT networks. Internet Things. 2024;26(71):101162. doi:10.1016/j.iot.2024.101162.
13. UNSW. The UNSW-NB15 dataset [Internet]. 2021 [cited 2026 Jan 1]. Available from: https://research.unsw.edu.au/projects/unsw-nb15-dataset.
14. Ali WA, Manasa KN, Aljunid M, Bendechache M, Sandhya P. Review of current machine learning approaches for anomaly detection in network traffic. J Telecommun Digit Econ. 2020;8(4):64–95. doi:10.18080/jtde.v8n4.307.
15. Fathima A, Khan A, Uddin MF, Waris MM, Ahmad S, Sanin C, et al. Performance evaluation and comparative analysis of machine learning models on the UNSW-NB15 dataset: a contemporary approach to cyber threat detection. Cybern Syst. 2025;56(8):1160–76. doi:10.1080/01969722.2023.2296246.
16. Sowmya T, Mary Anita EA. A comprehensive review of AI based intrusion detection system. Meas Sens. 2023;28(4):100827. doi:10.1016/j.measen.2023.100827.

17. Ofoegbu KDO, Osundare OS, Ike CS, Fakeyede OG, Ige AB. Real-time cybersecurity threat detection using machine learning and big data analytics: a comprehensive approach. Comput Sci IT Res J. 2023;4(3):478–501. doi:10.51594/csitrj.v4i3.1500.
18. Ofusori L, Bokaba T, Mhlongo S. Explainability and interpretability of artificial intelligence use in cybersecurity. Discov Comput. 2025;28(1):241. doi:10.1007/s10791-025-09760-6.
19. Almolhis NA. Intrusion detection using hybrid random forest and attention models and explainable AI visualization. J Internet Serv Inf Secur. 2025;15(1):371–84. doi:10.58346/jisis.2025.i1.024.
20. Patil S, Varadarajan V, Mazhar SM, Sahibzada A, Ahmed N, Sinha O, et al. Explainable artificial intelligence for intrusion detection system. Electronics. 2022;11(19):3079. doi:10.3390/electronics11193079.
21. Husain S. Deep learning-based AI attack detection: a real-world cybersecurity dataset approach. J Inf Syst Eng Manag. 2025;10(32s):98–106. doi:10.52783/jisem.v10i32s.5192.
22. Selvarajan P, Salman R, Ahamed S, Jayasuriya P. Networks intrusion detection using optimized hybrid network. In: 2023 International Conference on Smart Computing and Application (ICSCA); 2023 Feb 5–6; Hail, Saudi Arabia. p. 1–6. doi:10.1109/ICSCA57840.2023.10087611.
23. Said NMM, Ali SM, Shaik N, Begum KMJ, Shaban AA, Samuel BE. Analysis of Internet of Things to enhance security using artificial intelligence based algorithm. J Internet Serv Inf Secur. 2024;14(4):590–604. doi:10.58346/jisis.2024.i4.037.
24. Jagannathan J, Prasanna Kumara SG, Lakshmi Narayana T, Alam MS. Securing wireless communication using novel transfer learning for encryption in wireless networks. ICTACT J Commun Technol. 2023;14(3):3005–12. doi:10.21917/ijct.2023.0447.
25. Nowroozi E, Mohammadi M, Rahdari A, Taheri R, Conti M. A random deep feature selection approach to mitigate transferable adversarial attacks. IEEE Trans Netw Serv Manag. 2025;22(6):5301–10. doi:10.1109/TNSM.2025.3594253.
26. Fenjan A, Almashhadany MTM, Ahmed SR, Fadel HA, Sekhar R, Shah P, et al. Adaptive intrusion detection system using deep learning for network security. In: Proceedings of the Cognitive Models and Artificial Intelligence Conference; 2024 May 25–26; İstanbul, Turkiye. p. 279–84. doi:10.1145/3660853.3660928.
27. Ali G, Samuel A, Mijwil MM, Al-Mahzoum K, Sallam M, Olalekan Salau A, et al. Enhancing cybersecurity in smart education with deep learning and computer vision: a survey. Mesopotamian J Comput Sci. 2025;2025:115–58. doi:10.58496/mjcsc/2025/008.
28. Kikissagbe BR, Adda M. Machine learning-based intrusion detection methods in IoT systems: a comprehensive review. Electronics. 2024;13(18):3601. doi:10.3390/electronics13183601.