

Ensuring Network Security in Remote Work Environment

MSc Research Project
MSc in Cybersecurity

Mustafa Maveeya Maaz Mohammad
Student ID: x23213027

School of Computing
National College of Ireland

Supervisor: Kamil Mahajan

**National College of
Ireland MSc Project
Submission Sheet**



School of Computing

Student Name: Mustafa Maveeya Maaz Mohammad
Student ID: x23213027
Programme: MSc in Cybersecurity **Year:** 2024-25
Module: Practicum Part 2
Supervisor: Kamil Mahajan
Submission Due Date: 29/01/25
Project Title: Ensuring Network Security in Remote Work Environment
Word Count: 9195 **Page Count:** 22

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: Mustafa Maveeya Maaz Mohammad

Date: 29/01/25

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	<input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	<input type="checkbox"/>
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.	<input type="checkbox"/>

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):	

Ensuring Network Security in Remote Work Environment

Mustafa Maveeya Maaz Mohammad
x23213027

Abstract

The outbreak of the COVID-19 pandemic has led to the increased use of remote work, and this has greatly improved the uncertainties of cybersecurity hence the need for sophisticated IDS for remote systems. In this work, a comparative study is presented between IDS ML and DL models by using the CICIDS-2017 dataset. The analysis compares clustering algorithms – K-Means, DBSCAN, and DL models – LSTM, Attention LSTM, and Transformer – in threat identification, including zero-day threats. Among the models, the highest precision (95.78%) is seen for the Transformer, as well as a relatively high F1-score with a value of 93.76%. This ability of the Transformer model is connected to the model's ability to work with dependencies between features. LSTM and Attention LSTM models both provided good rates of recall at 98.19% and 98.15% respectively, which is especially good for identifying frequently occurring, known attack types. In this task, supervised methods outperformed unsupervised methods like KMeans and DBSCAN with 15.25% and 0.76% accuracy, respectively, as both algorithms are weak when it comes to a high number of attributes data. This discussion also highlights the use of DL model's effectiveness in structured IDS application while paving ways to consider the future research with transformer-based algorithms for zero-day attacks.

1 Introduction

The rise of remote work, catalyzed by global events such as the COVID-19 pandemic, has revolutionized workplace dynamics. However, this shift has also introduced significant cybersecurity challenges, with remote work environments often being prime targets for cyberattacks. According to the 2023 IBM Cost of a Data Breach Report, the average cost of a data breach has reached \$4.45 million globally, a 15% increase over three years, with breaches in remote work environments costing an additional \$1 million on average (IBM, 2024).

The transition to remote work has not only expanded the attack surface for malicious actors but also heightened the need for advanced intrusion detection systems (IDS) (Mallick and Nath, 2024; Mukherjee et al., 2024). Traditional IDS methodologies often struggle to adapt to dynamic, complex network environments characteristic of remote operations (Muneer et al., 2024). This research seeks to address these gaps through the development and evaluation of hybrid IDS frameworks leveraging machine learning (ML) and deep learning (DL) techniques.

There is new consideration in the cybersecurity environment because of more than 60% of organisational structures adopt the remote or hybrid working model (Tahmasebi, 2023). It is common to experience higher risks due to the use of insecure personal gadgets and devices, uncontrolled home networks, and no central IT control. Nevertheless, current IDS systems are mostly a type of reactive system, which requires that there must be a proactive form designed to prevent new threats from gaining ground.

This research is therefore informed by the current and emerging research gap of lack of IDS solutions for remote working environments, to support the effective identification of new emerging cyber threats.

In particular, the findings of this work are relevant to the field of cybersecurity, and more specifically, intrusion detection for remote workplace contexts. They give a performance based

review of IDS methodologies and their challenges. Based on these findings, this study proposes a suitable algorithm to enhance the detection accuracy and resilience. Moreover, it provides a benchmarking and validation of the proposed framework with other comparable approaches that lays the foundation for a valuable knowledge base in the field of cybersecurity with uses of ML and DL. These contributions are intended to help closing the gap between more traditional IDS models and today's demanding security environments of dispersed work contexts.

1.1 Research Aim and Objectives

The aim of this research is to develop a IDS framework optimized for remote work environments, by comparing state-of-the-art ML and DL techniques to improve detection accuracy and resilience.

Objectives:

- To conduct a comprehensive review of existing IDS methodologies, identifying limitations and opportunities for improvement.
- To implement and evaluate clustering techniques (e.g., K-means, DBSCAN) and DL architectures (e.g., LSTM, Attention LSTM, Transformer models) using CICIDS-2017 dataset.
- To compare the implemented models using Accuracy, Precision, Recall and F1-Score.

1.2 Research Question

This study is driven by the aim to obtain the answer to the following research question:

How effective are deep learning algorithms such as LSTM, Attention LSTM, and Transformer Models compared to the Unsupervised Kmeans and DBSCAN clustering algorithms in intrusion detection in developing a robust IDS in remote work environment?

Along with the question mentioned, the study will try to answer the given sub-questions.

Sub-questions:

- How effective are clustering-based methods like K-means and DBSCAN for anomaly detection in IDS?
- What are the comparative advantages of DL models, such as Attention LSTM and Transformer models, in intrusion detection?
- Which model will be the most suitable for detecting security threats in CICIDS-2017 dataset?

This work therefore fills the gap in the literature on cybersecurity by focusing on the issues arising from remote working conditions. Thus, it provides a comparative analysis of Unsupervised ML models and advanced DL models for IDS and the prevailing requirements of modern networks. Thus, the accomplishments achieved within this study could foster understanding of the proactive, adaptive IDS solutions that organizations can incorporate to prevent cyber threats effectively and, therefore, minimize their cost.

Rest of this thesis is organised into several key chapters to systematically address the research objectives. The Related Work chapter provides a comprehensive review of state-of-the-art intrusion detection system (IDS) methodologies. The Methodology chapter outlines the research design, datasets utilized, and the models employed in the study. In the Design Specifications chapter, the study design is detailed, including its components and architectural

structure. The Implementation chapter discusses the practical execution of the methodology, highlighting the tools, libraries, and workflows used. The Evaluation chapter presents an in-depth analysis of the experimental results. Finally, the Conclusion and Future Work chapter summarizes the key findings of the study, underscores its contributions, and identifies areas for further research and development.

2 Related Work

Thus IDS systems are vitally important for protection of remote work environment as more and more workers may become easy targets for cyber criminals the activity of which has increased in its sophistication. With the acceptance of work from home practices, the attack surface has been effectively enlarged and there is a need for new and quickly scalable IDS tools in order to addresses new and continually emerging threats. This chapter explains about the existing methodologies on intrusion detection and it is prepared based on the CICIDS2017 data set and other benchmarks. They also talk about clustering algorithms such as K-means and DBSCAN, about using deep learning methods like Attention LSTM and Transformer models, as well as about the importance of comparing such methods to define which ones are optimal. The subsequent sections provide post-literature review arguments for future improvement to the use of IDS in remotely working environments.

2.1 Recent Studies on CICIDS-2017 Dataset and Application of SMOTE for Class Balancing

Although there are many datasets freely available in the literature, CICIDS2017 has recently become significant for evaluating the performance of intrusion detection methodologies. It has been shown useful in assessing a variety of machine learning and deep learning algorithms.

Widodo et al. (2024) used SMOTE in dealing with class imbalance issue in the classification of CICIDS2017 dataset. The authors claimed their model has attained impressive performance: accuracy of 98.1%, and F1-score of 96.8%. They proved that SMOTE oversampled the minority classes with a greater percentage of recall and important infrequent but dangerous intrusions type. Likewise, Alfrhan et al. (2020) noted as the ensemble method aided by SMOTE enhanced the minority class recall from 72.3% to 94.8%, and AC to 96.7%. These results underscore the importance of class balance techniques in the performance of IDS.

Another reason has been identified by Khan et al. (2024) that has categorized dataset quality and features relevance as the critical factors of IDS. CICIDS2017 based comparison of different machine learning models to achieve overall detection rate of 94.6%, however they concluded that for making IDS more robust more amendments have to be done to the dataset.

Catillo et al. (2023) also investigated adversarial robustness in IDS models with CICIDS2017. Autoencoders also proved to be less vulnerable to adversarial attacks; the accuracy decreased to 95.2% from 98.3%, while decision trees, to 84.7% from 96.1%. These findings imply that a good and sound model is critical to be implemented in real life, particularly in circumstances such as working remotely where adversarial attacks are more common.

2.2 Application of K-means and DBSCAN in Intrusion Detection

Algorithms such as K-mean and DBSCAN have been used frequently in intrusion detection because of their capacity to perform learning from the unlabeled datasets.

Rashid et al (2024) developed another integrated model with K-means, DBSCAN, and SMO; the model has 95.7% detection accuracy and 2.3% false alarm probability. This work has shown that deep integration of clustering with classification techniques can enhance IDS performance, thus, calls for the design of hybrid models, to be extensively used in the IDS.

Mustafa & Husien (2023) proposed the adaptive DBSCAN-GWO model for RT-benet detection with increases the detection accuracy up to 98 %. Compared to only DBSCAN score from the original model (80%) and K-means (79.6%), this model proved that optimizing parameters among those clustering algorithms can benefit from parameters tweaking.

In Karoui & Abd (2024), the authors fused the K-means clustering algorithm with XGBoost for detection of new attacks. For detection rate, their model yielded 95.8% on a self-generated dataset and 94.3% on a Kaggle dataset showing that clustering approaches, when implemented together with supervised learning are feasible.

According to Deng et al. (2020) who fine-tuned DBSCAN for anomaly detection, it obtained a mean accuracy of 97.3%. Based on their results, they observed that the noise tolerance characteristics of DBSCAN would prove valuable for IDS in dynamic contexts like remote working circumstances.

2.3 Application of Deep Learning Techniques (Attention LSTM, LSTM, and Transformer Models)

Attention LSTM, LSTM, and Transformer models have been found to be suitable when identifies intrusion attacks with high precision.

Laghrissi et al. (2021a) also used LSTM for IDS with an accuracy level of 98.7% to the binary classification and 96.8% to multi-class. Use of PCA on the feature reduction improved the efficiency of the model while still maintaining a high accuracy.

The incorporation of attention mechanisms enhanced the LSTM and its behaviour as evidenced by Laghrissi et al. (2021b) where acheved an accuracy of 99.09% in a binary class problem. Yang et al. (2020) introduced Attention together with LSTM to boost feature importance handling; the accuracy was 98.2% and the false positive rate was 1.8% lower.

Transformer models have recently been used in IDS because of their enhanced contextual training process. On this basis Rahali & Akhloufi = (2021) proposed a new named MALBERT which is a BERT based model with binary as well as multi-class classification of 99.2% and 97.4%, separately. Kheddara (2024) reported in his survey on using Transformers such as BERT and Vision Transformers (ViT) for IDS where they proposed models that yielded accuracy of up to 99.0% on CICIDS2017.

Proposed by Fu et al. (2022), the integration of attention mechanisms into Bi-LSTM also solved the data imbalance problem, and it yielded 90.73% accuracy and an F1-score 89.65%. These results show how deep learning can help overcome complicated problems of intrusion in remote environments.

2.4 Need for Comparative Analysis of these Techniques

The nature and variety of the described approaches call for a comparative evaluation of the methods most suitable for intrusion detection. Even though simple methods such as K-means and DBSCAN are useful in detecting outliers or anomalies, introducing methods from supervised learning to these methods provides exceptional results. In particular, when it comes to the choice in favour of deep learning models paying attention and Transformers, greater accuracy is achieved at the price of increased computational measures.

Maseer et al. (2021) comparatively analyzed 10 traditional, MLP, and eight DL models to analyze CICIDS2017 dataset – and found that accuracy of Decision Trees was the highest and 99.75% for CICIDS2017. Nonetheless, Naive Bayes was more accurate in specific attacks suggesting that the IDS should combine the model that best fits the particular system requirements.

LSTM, the Attention LSTM and the Transformer models are another level more in-depth showing the comparison of high accuracy, cheap interpretability and cheap computation. For example, better than standalone LSTM, ‘Attention LSTM’ rises to the next level of competence of ‘Transformers’ which give higher meaningful context sense making of IDS for diversified complex remote work environment situations.

2.5 Conclusion

A detailed assessment of intrusion detection systems (IDS) based on clustering and deep learning mechanisms emerges from the reviewed literature in this chapter. An in-depth check of these particular strategies together with their performance against zero-day assaults would strengthen our grasp of their real-world applicability. Each individual method including LSTM, Attention LSTM, Transformer models, K-Means and DBSCAN goes into deep analysis yet research typically skips essential direct evaluations of these methods' problem areas and strengths. Participants need to understand exactly how every method operates against unidentified threat patterns when conditions transform in today's evolving zero-day threat scenario. A performance-focused analysis between techniques leads to more transparent results about how new cyber vulnerabilities are addressed rather than an exploration of unified methods.

This chapter has presented a critical analysis of recent advances in intrusion detection with regards to the CICIDS2017 dataset, clustering algorithms and deep learning frameworks. The results highlight the need to adopt measures to counter class imbalance, employ the tactics of both machine learning and deep learning, and adopt deep learning tactics to improve IDS effectiveness. Nevertheless, the call for a more exhaustive comparative study persists because such a study would offer a cleaner picture of the advantages and appropriateness of the techniques in certain applications. This knowledge is important in order to design effective IDS that will be able to fully address the needs and issues of remote work scenarios.

3 Methodology

The methodology adopted in this study is tailored to achieve the dual objectives of identifying the most suitable model for IoT network traffic classification and evaluating the zero-day attack detection capabilities of unsupervised models. A systematic approach was undertaken, encompassing data acquisition, preprocessing, exploratory data analysis (EDA), feature engineering, model selection, and evaluation. At every step, the methodology was chosen to address specific challenges, such as class imbalance, high dimensionality, and the need for robust evaluation metrics, ensuring that the results are reliable and generalisable (Ullah & Mahmoud, 2021; Alsoufi et al., 2021).

Figure 1 below depicts the general flow of the methodology incorporated in the study.

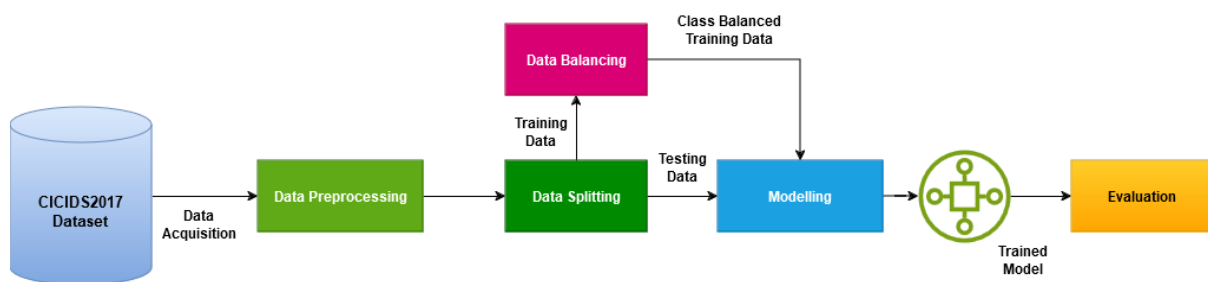


Figure 1: Methodology Flow of the Study

The different sections of the methodology are discussed in detail below:

3.1 Data Acquisition

For this study, the CICIDS2017 dataset that has been known to provide realistic emulation of IoT network traffic was employed. The attack instances contain variety of attack types such as DDoS, Port Scans, Infiltration and benign instances, making it suitable for a comparative assessment of classification models. Thus, employing datasets within the collection for instance,

- Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv
- Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX.csv
- Friday-WorkingHours-Morning.pcap_ISCX.csv
- Monday-WorkingHours.pcap_ISCX.csv
- Thursday-WorkingHours-Afternoon-Infiltration.pcap_ISCX.csv
- Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv
- Tuesday-WorkingHours.pcap_ISCX.csv
- Wednesday-workingHours.pcap_ISCX.csv

offered encompassing evaluation of multiple traffic scenes guaranteeing solidity. This is why the choice of CICIDS2017 can be explained, given that it has been used in cybersecurity research for designing and testing IDSs (Ullah & Mahmoud, 2021).




Label	
BENIGN	2273097
DoS Hulk	231073
PortScan	158930
DDoS	128027
DoS GoldenEye	10293
FTP-Patator	7938
SSH-Patator	5897
DoS slowloris	5796
DoS Slowhttptest	5499
Bot	1966
Web Attack  Brute Force	1507
Web Attack  XSS	652
Infiltration	36
Web Attack  Sql Injection	21
Heartbleed	11
Name: count, dtype: int64	

Figure 2: Different Attacks Present in the Dataset

The values of common columns between the datasets were kept in order to be able to merge all into a single DataFrame. To handle missing data and redundant or duplicated columns, a process of data cleaning by removing the duplicated rows and by mean imputation of missing values were carried out and applied to clean the data. These preprocessing steps made it possible to work on a clean dataset once we had achieved the next level and also made sure that the properties of distribution were conserved (Little & Rubin, 2002).

3.2 Exploratory Data Analysis (EDA)

EDA was conducted to understand the underlying structure of the dataset and inform subsequent modelling decisions. Count plots highlighted significant class imbalances, particularly between benign and malicious traffic, which could adversely affect the performance of classification models. This insight justified the subsequent application of SMOTE for balancing the dataset.

Scatter plots and violin plots revealed relationships between key features, such as Flow Duration and Total Fwd Packets, and their distributions across traffic types.

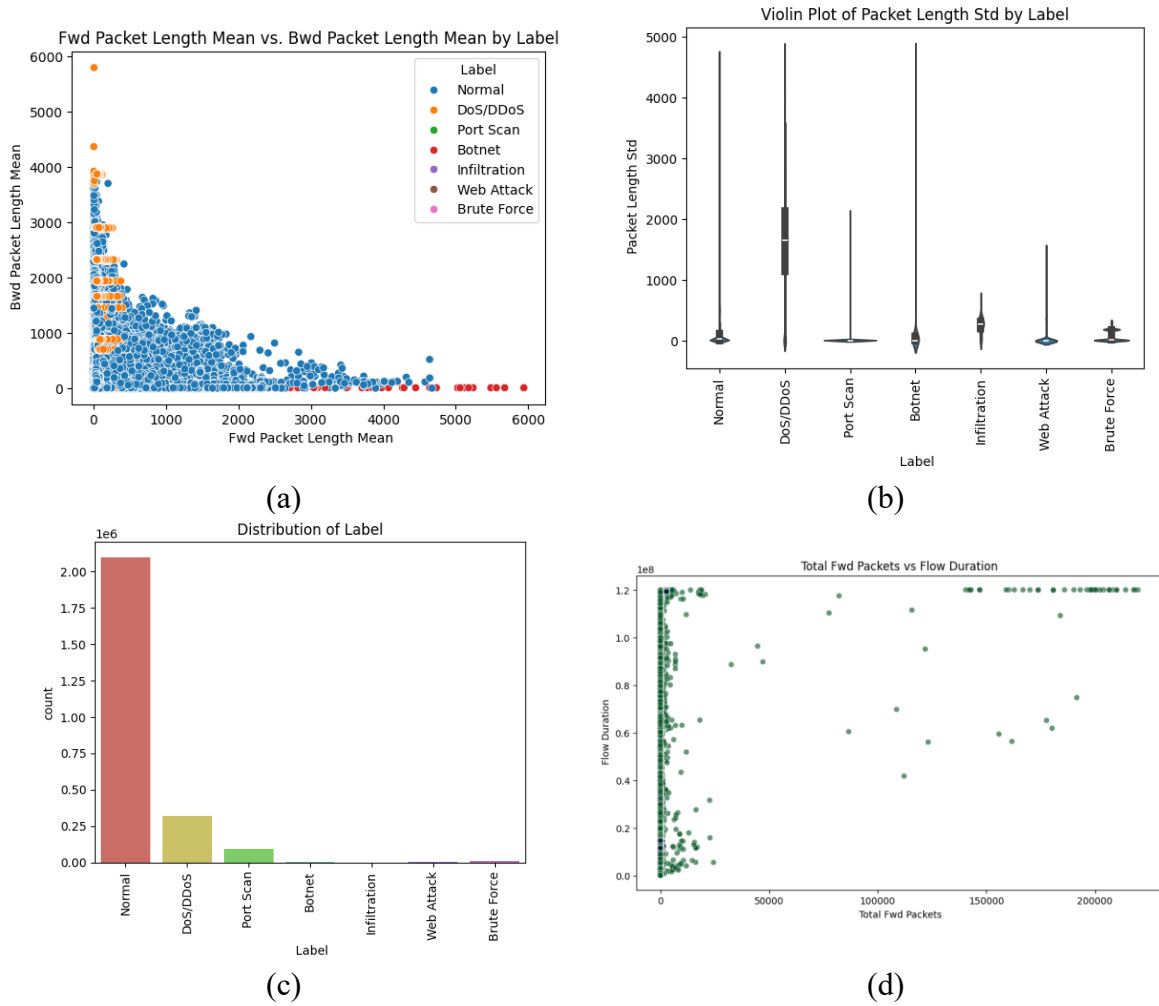


Figure 3: Exploration and Visualisation of the Dataset

Correlation heatmaps were also employed to identify multicollinearity, guiding the removal of highly correlated features, which is critical to improving model interpretability and performance (Dormann et al., 2013). These steps ensured that the dataset was both comprehensive and optimally structured for analysis.

3.3 Data Preprocessing

The rationale for preprocessing was to deal with the issues of class imbalance, high dimensionality, along with feature scaling so as to provide a good set up for the decided models. Since business intelligence involves comparison of class labels, classes had to be encoded into numerical representation and labels and sub-labels such as HTTP floods and ICMP floods had to be generalised and combined into a single field such as DoS/DDoS. Features were scaled to the mean of zero and standard deviation of one using StandardScaler because different features may be measured on different scales, and this is a big problem for those classification algorithms whose performance can be affected by the ranges of the features such as the Neural Networks (Hastie et al., 2009). Computation and noise issues were resolved by using Principal Component Analysis (PCA) where the first two components retained 95% of the variance (Jolliffe & Cadima, 2016) for generalisation. In addition to the handling of ordinal features noted in the exploratory phase, SMOTE was used in handling class imbalances because SMOTE provides samples for underrepresented classes and to reduce the bias towards majority classes, which will generally improve the classifiers' efficiency for imbalanced datasets (Chawla et al., 2002).

3.4 Feature Engineering

Feature engineering brought more value to effectively represent the dataset by cleaning and reorganizing some of the data to capture more complex relationships. Features with a correlation coefficient greater than 0.9 were removed, reducing multicollinearity and improving computational efficiency. Since EDA is a simple and intuitive way of exploring the data, new features were designed with respect to the specific theoretical domain and the outcome of the EDA, for example, interaction terms and statistical summaries. This process was meant to prepare the dataset for the chosen models to make sure all the features that would be included were as useful as possible.

3.5 Modelling

To evaluate both, the classification accuracy, as well the capacity of the algorithm to learn new categories of attack not seen before during training, the work includes both supervised and unsupervised models only. These models have for been selected due to the potential of processing high dimensionality and complex data set in the context of cybersecurity (Alsoufi et al., 2021).

1. Supervised Models:

- LSTM: Long Short-Term Memory networks were selected because they can effectively use network traffic data, which include temporal dependencies in order to distinguish between normal and malicious behaviour (Hochreiter and Schmidhuber, 1997).
- Attention LSTM: An Enhanced Neural Network model built under LSTM to attend the critical features that have high relevance while strengthening the interpretation and execution ability (Vaswani et al., 2017).
- Transformer: This model uses the self-attention technique for sequence modeling that is efficient and especially suitable when working with high-dimensional input data sets (Devlin et al., 2019).

2. Unsupervised Models:

- K-Means: used to group similar data points together, bearing in mind the effectiveness of the algorithm in identifying anomalous patterns associated with zero-day attack. This technique is simple and efficient, and therefore it is common in clustering (Lloyd, 1982).
- DBSCAN: This kind of clustering algorithm detects points to be clustered to create dense groups of data and also quickly defines all outliers which makes it good for usage when detecting noisy data or potential zero day attacks (Ester et al., 1996).

The use of both supervised and unsupervised ones made the given methodology guarantee the assessment of the classification effectiveness and the characteristics of detecting new previously unknown threats.

3.6 Training and Evaluation

This training process aimed at giving proper comparison between the models to come up with a clear result. The total sample data was divided into training and testing set of 80:20 proportion, which maintained proportional class distribution. Validation was used for all the considered supervised models to prevent the model from overfitting during the process and to estimate the model's performance with high accuracy (Kohavi, 1995).

Evaluation metrics used were accuracy, precision, recall, F1-score and ROC-AUC to ensure, that all aspects of the model is taken into consideration. To assess the unsupervised models, confusion matrices were constructed to recognize particular misclassification patterns.

3.7 Summary

The proposed methodology was developed to meet the main problems concerning the IoT network traffic classification and the detection of new unauthorized attempts. The integration of SMOTE to tackle the problem of class imbalance, utilization of PCA to improve the feature selection process, and the application of robust evaluation metrics enhances the achievement of the study goals significantly. The usage of both supervised models focusing on the classification of known attacks and unsupervised models designed for estimating the capability of detection of zero-day attacks let for more thorough analysis. These choices were based on the knowledge gained from the literature review and the best practices in the cybersecurity field, which made the findings more valid and relevant.

4 Design Specifications

The specifics of the current study's design are oriented toward providing a more comprehensive comparison of the supervised and unsupervised models in IoT network traffic classification. And the objective here would be to assess the potential of these approaches in identifying Network Anomalies, especially Zero Day Attacks, using their parameters such as accuracy, precision, recall rate, and F1-score. Some of the components of the system design include the following: Various issues including the class imbalance problem, high dimensionality, among others are the main challenges that are catered for by the components of the system design.

4.1 System Architecture

Figure below depicts the system architecture employed in the implementation of the study.

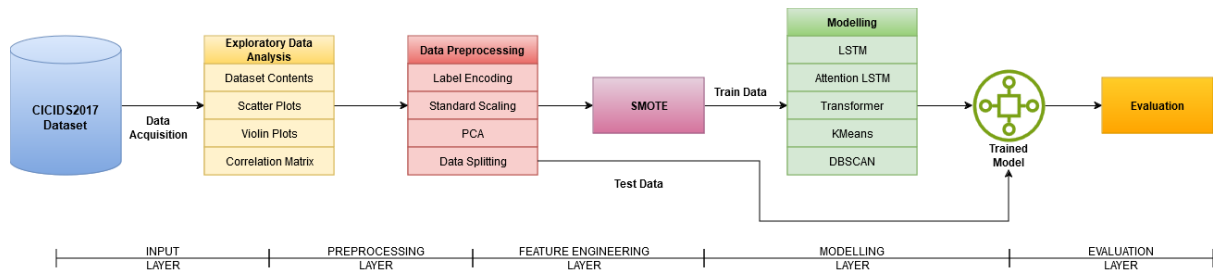


Figure 4: System Architecture

The system is structured into modular layers, with each layer performing a specific role to ensure seamless data processing and model evaluation:

4.1.1 Input Layer:

The input consists of the unprocessed information of network traffic in the CICIDS2017 dataset. The features in this dataset include the following; Packet Length Variance, Flow Duration, and Average Packet Size which encompass both legitimate and illegitimate network activities. The selection of this dataset guarantees that the different models can indeed be tested against a real and complex IoT network traffic.

4.1.2 Preprocessing Layer:

Data Cleaning and Imputation: Before data analysis, all the incomplete records with missing values were imputed by mean so as to maintain the sample statistics of the study (Little & Rubin, 2002).

Standardisation: Features were normalized with StandardScaler of scikit-learn to achieve uniformity of data magnitudes because feature ranges significantly affect algorithms like neural networks (Hastie et al., 2009).

Feature Selection: Features with high correlation were deleted to reduce overlapping and complexity of interpretative evaluations thus enhancing model effectiveness.

Combined with these steps, they made a dataset as suitable as possible for modeling, with reference to the problems of inconsistency, multicollinearity, and variability of features 'scales.

4.1.3 Feature Engineering Layer:

Dimensionality Reduction: PCA was applied to reduce the number of features while retaining 95% of the variance, addressing computational challenges and improving model generalisation by removing noise (Jolliffe & Cadima, 2016).

Class Balancing: The Synthetic Minority Oversampling Technique (SMOTE) was employed to generate synthetic samples for minority classes, reducing the class imbalance observed during exploratory analysis. This ensured that the models were not biased toward majority classes, improving their overall reliability (Chawla et al., 2002).

4.1.4 Model Layer:

This layer evaluates the dataset using multiple models, chosen for their specific strengths in handling IoT traffic data. Supervised models classify known traffic patterns, while unsupervised models assess clustering capabilities for detecting zero-day attacks.

4.1.5 Evaluation Layer:

This layer produces predictions, evaluates model performance using metrics such as accuracy, precision, recall, and F1-score, and visualises results through confusion matrices and ROC curves. The outputs provide a comprehensive view of the comparative performance of the models.

4.2 Models Used

4.2.1 Supervised Models

Long Short-Term Memory (LSTM): LSTMs are a specialised type of Recurrent Neural Network (RNN) which are designed to process sequential data by addressing the vanishing gradient problem inherent in RNNs. These models are particularly effective in capturing temporal dependencies in IoT traffic, such as behavioural patterns over time. Various gates in the

- **Forget Gate:** Controls how much information from the previous cell state c_{t-1} should be retained:

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

Equation 1

- **Input Gate:** Determines which new information to update in the cell state:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

Equation 2

- **Cell State Update:** Combines the forget gate and input gate outputs:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$$

Equation 3

- **Output Gate:** Produces the final hidden state h_{t_t} for the current timestep:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t \odot \tanh(c_t)$$

Equation 4

Here, σ is the sigmoid activation function, \tanh is the hyperbolic tangent function, \odot is element-wise multiplication, and W and b are learnable weights and biases.

Attention LSTM: The Attention LSTM extends the LSTM model by incorporating an attention mechanism that prioritises important features or timesteps, improving classification accuracy and interpretability.

- **Alignment Score:** Determines the relevance of each hidden state h_i to the current timestep t :

$$e_{ti} = h_t^\top W_a h_i$$

Equation 5

- **Attention Weights:** Normalises the alignment scores using softmax:

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_j \exp(e_{tj})}$$

Equation 6

- **Context Vector:** Computes a weighted sum of hidden states based on attention weights:

$$c_t = \sum_i \alpha_{ti} h_i$$

Equation 7

The context vector c_t enhances the LSTM's ability to focus on critical features, improving its effectiveness in complex traffic classification tasks (Vaswani et al., 2017).

Transformer: The Transformer model is a sequence-processing architecture that uses self-attention mechanisms instead of recurrent connections. This design enables efficient processing of high-dimensional data and captures long-range dependencies.

Self-Attention: Computes weighted dependencies between all input tokens:

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^\top}{\sqrt{d_k}} \right) V$$

Equation 8

Here, Q , K , and V are the Query, Key, and Value matrices, and d_k is the dimensionality of the key.

Multi-Head Attention: Enhances the model's ability to focus on different parts of the sequence simultaneously:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W_O$$

Equation 9

Where:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Equation 10

By using self-attention and multi-head attention mechanisms, the Transformer excels in tasks requiring scalability and interpretability (Devlin et al., 2019).

The Query or Q matrix illustrates the “current focus” paradigm of the model in relation to an input. In relation to IoT traffic data it may reflect a particular aspect of a traffic sample, for instance Packet Length Variance or Flow Duration which the model is attempting to evaluate against other aspects. The Key (K) matrix serves as a comparison of the Query or other texts. In connection with this, there are other aspects or meters of the input data which play significant roles in determining the Key with reference to the importance of the Query, and all these are collectively known as the Key matrix. For instance, it might stand for other traffic samples' aggregate characteristics within the latter. The last two columns of the Value (V) matrix hold the information that should be summed or extracted depending on the attention scores calculated between the Query and the Key. In IoT traffic analysis, the Value matrix could be interpreted as a weight of some features such as corresponding to a particular attack type or a benign network activity. These two matrices together help the model to check relations, filter/move important features & enhance classification competence and efficiency.

4.2.2 Unsupervised Models

Unsupervised models were included to assess their capability in detecting zero-day attacks, where labelled data is unavailable. These models identify clusters and anomalies based on feature similarities.

K-Means Clustering: K-Means is a centroid-based clustering algorithm that iteratively partitions the dataset into k clusters. The objective function minimises intra-cluster variance:

$$J = \sum_{i=1}^k \sum_{x \in C_i} \|x - \mu_i\|^2$$

Equation 11

Where:

- k: Number of clusters.
- C_i : Cluster i.
- μ_i : Centroid of cluster i.
- $\|x - \mu_i\|^2$: Squared Euclidean distance between a point and its cluster centroid.

Centroids are updated iteratively as:

$$\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$$

Equation 12

K-Means is effective for large, structured datasets but assumes spherical cluster shapes, which may limit its performance for complex IoT traffic data.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise): DBSCAN identifies clusters based on density, classifying sparse points as noise. This capability makes it particularly robust for detecting anomalies in non-linear datasets.

- **Core Point:** A point p is a core point if:

$$|\text{Neighborhood}_\epsilon(p)| \geq \text{MinPts}$$

Equation 13

- **Directly Density-Reachable:** A point q is directly density-reachable from p if:

$$\|p - q\| \leq \epsilon$$

Equation 14

- **Cluster Formation:** Clusters are formed by connecting core points and their density-reachable neighbours.

DBSCAN does not require a predefined number of clusters and is robust against noise, making it a powerful tool for identifying potential zero-day attacks in IoT traffic (Ester et al., 1996).

4.2.3 Summary

In this work, the use of both supervised and unsupervised models allows for a thorough assessment of IoT traffic classification and anomaly detection. For the labelled data and for the temporal data where internal dependencies are significant, models like LSTM, Attention LSTM and transformer have high efficiency and, for the zero-day attacks, clustering models like K-means and DBSCAN are highly efficient. The specific formulas which make up these models are presented to emphasize the theoretical aspect of the constructs and to demonstrate how they fit into the comparative structure of the study.

5 Implementation

The procedures described in this chapter include creating the environment, preparing the dataset, training the models and the evaluation done. This phase needed the deployment of sophisticated tools and methods to apply rigor, flexibility, and the capacity to conduct a thorough comparative assessment of supervised and unsupervised techniques. The goal of the study was to determine the best performing models for classifying IoT network traffic and to test the effectiveness of the unsupervised models for the detection of previously unseen-attack types.

Data preprocessing was the first step that was followed in the process of implementing the approach. Therefore, CICIDS2017 dataset collected from public repositories faced issues like missing values, class imbalance and high dimensions and to overcome all these issues data preprocessing was done. Preprocessing of features was done with scaling methods For dimensionality reduction, Primary component analysis was used . Indeed, there was an

imbalance of classes in the datasets, thus, in order to balance the classes Synthetic Minority Oversampling Technique (SMOTE) was used to create synthetic samples. Moreover, exploratory data analysis helped identify the relevant features and their relevance to each other in the decision-making regarding model choice.

5.1 System Configuration

The implementation relied on a robust hardware and software setup to handle the computational demands of deep learning and clustering models. The configuration details are as follows:

- **Hardware Specifications:**
 - RAM: 8GB DDR4
 - Storage: 1TB HDD + 256GB NVMe SSD
 - GPU: NVIDIA GTX 1650 (4GB GDDR5 Graphics)
 - Processor: Intel Core i5, 9th Generation
- **Operating System:**
 - Windows 11 (64-bit)

The dedicated GPU (NVIDIA GTX 1650) significantly accelerated the training of deep learning models, particularly those using architectures like Long Short-Term Memory (LSTM) and Attention LSTM. The NVMe SSD enhanced the data read/write speed, optimising the overall preprocessing and training pipeline.

5.2 Development Configuration

The development environment was tailored to support efficient implementation and testing of machine learning workflows. Key tools and libraries included:

- **Programming Language:** Python 3.9
- **Integrated Development Environment (IDE):** Jupyter Notebook (via Anaconda distribution)
- **Libraries:**
 - Data Handling and Analysis: NumPy, Pandas
 - Visualisation: Matplotlib, Seaborn
 - Machine Learning: Scikit-learn
 - Deep Learning: TensorFlow, Keras
 - Clustering and Sampling: SMOTE, DBSCAN, K-Means
 - Dimensionality Reduction: PCA, SciPy

These tools provided a seamless and modular framework for implementing data preprocessing, feature engineering, model training, and evaluation.

5.3 Hyperparameter Optimisations

In this study, hyperparameter optimisation depended on a manual search strategy utilizing parameters like learning rate, batch size, configuration of the hidden layers and dropout rates. The technique employed here was to adjust just one parameter at a time, record changes in key performance metrics (accuracy, precision and recall), and account for the optimally married set of parameters that yields the best results. As these shifts were monitored closely, it was possible to detect the subtleties of the interaction between parameters and to find the conditions in which the predictive quality worsens or improves. While this involved some amount of trial and error,

it was the best option compared against more automated strategies, grid search or Bayesian optimisation, which can be computationally costly and time consuming. Manual tuning also had the advantage of being hands on and instant in terms of responding to unexpected behaviour, and adjusting parameters as close as possible to observed patterns in the data. What's more, this iterative approach allowed me to understand model capacity and overfitting tendencies in fine grain, enabling me to strike a reasonable tension between complexity and generalisability. By taking a manual and manual first approach to fine tuning hyper parameters, I was able to slowly glean a lot deeper insights into how each hyperparameter ended up affecting performance overall and ultimately ended up with a much more tailored and efficient way in which to configure each model.

5.4 Model Implementation

5.4.1 Long Short-Term Memory (LSTM)

In this study, an LSTM model was utilized to understand temporal features of the IoT network traffic data. This architecture was well-suited for the sequential characteristics of network traffic because LSTMs are designed to handle time-series data well (Hochreiter & Schmidhuber, 1997). The model architecture included:

- An LSTM layer with 50 units and ReLU activation: This configuration enabled the model to capture long term dependencies in the input sequence which is essential in differentiating normal traffic behaviours or cyber-attacks.
- A Dropout layer: The last stage was a form of connecting layers' regularisations by using dropout in order to prevent overfitting, which may arise from high number of parameters through a large number of layers in deep learning architectures.
- A Dense output layer with softmax activation: This helped in the multi-class classification by weighting the class and assigning some probability to it.

The model was compiled using the Adam optimiser that is known for its ability to train deep learning models and binary cross-entropy loss function that is used for binary and multi-class classification. The proposed LSTM model was trained with 10 epochs and a batch size of 128 and proved the ability to capture patterns of traffic data, based on accuracy and recall metrics. The rationale for selecting LSTM goes as follows: LSTM can operate on sequential data and eliminate vanishing gradient problems, which are crucial for analyzing IoT networks (Goodfellow et al., 2016).

5.4.2 Attention LSTM

As part of enhancing the performance of the LSTM model which is a sequential model, an attention mechanism was incorporated. The attention mechanism weighed the crucial significance of the features in the input data for the model during the course of training. The architecture included:

- An LSTM layer with return sequences enabled: A change was made to the attention mechanism to enable the processing of the output sequences, to determine feature relevance at each time.
- A custom Attention layer: To explain the enhanced performance of this layer, it determined attention scores which helped to prioritize the most important features.
- A Dropout layer: L2 regularisation, moreover, was continued to ensure that estimates do not overly fit models, especially when implemented in deep networks.
- A Dense output layer with sigmoid activation: This enabled binary classification by probability in last layer which supported this layer.

The inclusion of attention mechanisms provided the model with higher-level performance especially for those datasets with a large number of features in which specific features (such as , anomalous packet lengths) were much more informative about the given attack types compared to others. The adoption of attention is because the authors observed that it improved model focus and interpretation, particularly where detailed feature prioritization is critical (Vaswani et al., 2017).

5.4.3 *Transformer Model*

To handle sequential data of high dimensionality, Transformer model was used, which uses self-attention for capturing dependencies existing between any timesteps of input data. This model was chosen for its capacity to scale up and its computational speed, as it lacks recurrent connections unlike LSTMs do. The architecture comprised:

- A multi-head self-attention layer: This component made it possible for the model to focus on various features in the same input sequence while acknowledging the intricate dependencies found between them.
- Dropout and Layer Normalisation layers: These layers provided the trained model a robust preprocessing to check the problem of overfitting during training.
- A Global Average Pooling layer: A lower dimensionality was obtained by pooling the feature information so that computation was simplified before the output layer.

The Transformer model was selected due to its capacity for working with complex feature interdependencies and its high performance in the high-dimensional environment. Different from recurrent models, the Transformers have a comparative advantage in capturing both local and global contextual information necessary for IoT traffic classification tasks (Devlin et al., 2019).

5.4.4 *K-Means*

This unsupervised approach was configured with a predefined number of clusters matching the dataset's unique class labels. While effective for structured datasets, K-Means assumes spherical cluster shapes, which may limit its application to non-linear patterns in IoT data. The choice of K-Means was justified by its simplicity and efficiency in detecting broad clustering patterns (Lloyd, 1982).

5.4.5 *DBSCAN*

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) was deployed to evaluate its capability in detecting zero-day attacks. Unlike K-Means, DBSCAN identifies clusters based on density rather than distance, making it robust against noise and capable of handling non-linear distributions. The algorithm's key parameters included:

- **ϵ** : The neighbourhood radius, optimised to capture dense regions.
- **min_samples**: The minimum number of points required to form a cluster.

DBSCAN's clustering process is governed by the following principle:

- For a point p , if the number of points within ϵ -radius (including p) is greater than or equal to **min_samples**, p is considered a core point, forming a cluster.
- Points within ϵ -radius of a core point belong to the cluster; other points are classified as noise.

DBSCAN was justified by its strength in identifying anomalies and its flexibility in clustering irregular patterns, critical for detecting previously unseen attack types in IoT datasets (Ester et al., 1996).

5.5 Deployment Scenarios

Although the trained ML and DL models performed well in controlled tests, applying them to remote-work settings requires careful planning and management. For example, they can be run on cloud or edge systems to analyse network traffic in real time. It is also crucial that security analysts understand how the models make decisions, so they can respond quickly and correctly if threats appear. By using these deployment methods and monitoring them continuously, the suggested intrusion detection solutions offer strong protection against the complex risks that remote-work environments often face.

5.6 Summary

In the implementation phase, it was possible to experiment with state of the art supervised and unsupervised models to classify IoT network traffic and assess anomaly detection. This study had employed some advanced assessment tools and work environment as well as demanding assessment criteria that make the findings valid and portable. Further conclusions will be drawn in the subsequent chapters where the detailed performance evaluation of each of the models developed will be described.

6 Evaluation

Evaluation chapter provides the analysis of the models used in the learning process within the framework of the study. The objective measures used are accuracy, precision, recall, F1 score, and confusion matrices. With regards to each model, their performance in relation to the context of IoT network traffic classification as well as their proficiency in detecting anomalous traffic which includes zero-day attack was examined.

6.1 LSTM Model

The proposed LSTM model got an average accuracy of 95.16%, which clearly showed its ability to capture temporal dependencies in sequential process data. Accuracy of the model is 57.69% and the recall of model is 98.19% that depict the capacity of the model to learn the pattern present in the network traffic while the F1 score of 64.41% explain moderate difficulties detected in striking a balance between precision and recall. Overall, the confusion matrix presents good performance in identifying the most frequent classes, but misclassification was observed in some instances of classes with fewer samples.

This is so because the LSTM model has a recurrent architecture which makes it efficient in processing sequences and retains temporal context. However, it uses all the features in an equal manner which may hinder it from being selective of the features to emphasize because the precision reduced slightly.

6.2 Attention LSTM

The attention-enhanced LSTM was found to enhance interpretability by compromising on irrelevant features during the training phase resulted in an accuracy of 94.56%. Slightly inferior to the plain LSTM in terms of accuracy (73.88%), but they performed equally well in terms of the recall (98.15%) the same has a similar F1 score 63.56%. The model was especially valuable at detecting low contrast attack patterns because an added attention mechanism enabled the system to selectively pay attention to important features of the input.

This is slightly lower than the full model, plain LSTM, which again poses the possibility of fine-tuning the attention weights to further prevent false alarms.

6.3 Transformer

The Transformer model gave an accuracy of 92.66% and precision of 95.78% the highest among all the supervised models. The analysis proved that all the metrics are balanced with the

recall at 92.66 % and F1 score at 93.76%. It indicates the confusion matrix the higher level of correct classification, and especially in complicated or minority classes.

The Transformer model is able to model high-order features due to the self-attention technique breaking up all dependencies without the sequential concern. On the same note, scalability and computational benefits enhance its applicability on large IoT datasets albeit, slightly lower recall compared to LSTM models indicates existence of gains in detecting minority classes.

6.4 KMeans

K-Means clustering achieved an accuracy of 15.25%, reflecting its limitations in handling the non-linear and high-dimensional nature of IoT traffic data. The precision of 74.81% and F1 score of 24.70% suggest that while the model could correctly classify certain samples, its overall effectiveness was hindered by its assumption of spherical cluster shapes. The confusion matrix indicates significant misclassification, particularly in detecting minority classes.

K-Means is effective for structured datasets but struggled in this context due to the complex interdependencies between features, highlighting its unsuitability for this specific application.

6.5 DBSCAN

DBSCAN, designed for anomaly detection, achieved an accuracy of 0.76%, with a precision of 83.35% and an F1 score of 1.45%. Its low recall (0.76%) reflects its inability to generalise effectively to the dataset, primarily due to the density-based clustering approach failing to form meaningful clusters in this high-dimensional space. The confusion matrix reveals that most samples were misclassified, with many labelled as noise.

While DBSCAN is robust against noise and effective for non-linear distributions, its performance in this study indicates that fine-tuning the hyperparameters (ϵ and min_samples) may be necessary for improved clustering and anomaly detection.

6.6 Discussion

The evaluation results present the peculiarities of the models compared in the given study, emphasizing their benefits and drawbacks, and, thereby, shedding light on the usability of this class of models for IoT network traffic classification. Supervised model preferred was the LSTM model as well as the LSTM model with attention that was able to recall their learned samples well enough and preferred for nicking known forms of attacks from the dataset. To avoid the arbitrary character of the problem, the LSTM allowed for stable and highly accurate reidentification of behavioral abnormalities in the traffic. While the LSTM achieved an overall capability for this task, attention-enhanced LSTM refined this by focusing on critical features in a dynamic manner, and was easier to interpret and fine tuned to provide detailed analysis on the given data. Still, the fact is that both of more complex LSTM variants with better recall rates show rather moderate precision which may mean the problem of frequent false positives and can be solved by further regularization or combination with other methods.

Model	Accuracy	Precision (Weighted)	Recall (Weighted)	F1-Score (Weighted)
Attention LSTM	0.9456	0.5618	0.9815	0.6356
LSTM	0.9516	0.5769	0.9819	0.6441
Transformer	0.9266	0.9578	0.9266	0.9376
K-Means	0.1526	0.7481	0.1526	0.2470
DBSCAN	0.0076	0.8335	0.0076	0.0145

Table 1: Model Evaluations

The Transformer model was among the top performers of both groups of methods, demonstrating high precision and similar results for all of the metrics. By breaking down the sequence into items and parallelly computing self-attention, it was capable of handling complex sequential data with feature interaction without recurrent architecture limitations. This was especially helpful as the Transformer had to deal with intertwined relationships within IoT traffic data. The trade off between precision and recall within the Transformer's result suggests that such architecture is fit for deployment in instances where both recall with high specificities is desired besides accurate detection.

On the other hand, the non-supervised clustering models; K-Means and DBSCAN fared poorly with the IoT dataset. K-Means clustering algorithm, though efficient for mathematical data with well defined, defined variable range and simple data structure, was not ideal for IoT traffic pattern data, primarily because of spherical assumption of the clusters. For instance, DBSCAN, which has been considered efficient in cases of noise and enough even in anomalous recognition scenarios of not linear data, has had its shortcomings due to the high-dimensionality as well as the sparseness of the data. That is why, it cannot form any important clusters and classify most of the samples, which proves the major challenges for clustering methods for zero-day attack detection. Nevertheless, the performance evaluation including unsupervised models gives more insights on the difficulty that might be incurred with anomaly detection in large networked environment and underlines the need for better improved clustering algorithm that tackles on the specific type of data.

These results highlight the advantages of employing supervised models that perform excellently in structured classification problems while arguing that there still is a large amount of work to be done to invent heuristic unsupervised solutions specifically for the security problem of IoT networks, where many zero-day attacks originate from.

7 Conclusion and Future Work

The proposed scheme was effectively used to analyse the efficiency of supervised and unsupervised models for intrusion detection capable of recognising the IoT network traffic in relation to remote work environments. The performances have revealed the merits and demerits of the models and paved the way to understanding their viability to be implemented in classifying network traffic and to detect anomalies.

Each participant presented recognised models, especially LSTM and the more advanced Attention Layer, which helped in identifying familiar attack patterns. In terms of recall, LSTM was higher, making it appropriate for recognizing frequent traffic behaviours, while the attention LSTM provided for increased interpretability due to feature spotlighting. The Transformer model was most accurate and yielded equally good results across all metrics, revealing its ability to handle high dimensional data with dependent features. These results endorse the proposed deep learning methods as optimal in structured classification applications in IoT systems.

On the other hand, the unsupervised models; K-Means and DBSCAN encountered some difficulties on account of IoT network traffic that proved to be high-dimensionality and non-linear in nature. K-Means was very inaccurate due to its' assumption that clusters are spherical. Despite that DBSCAN is more noise-tolerant than other clustering algorithms, it did not find significant clusters in the high dimensionality of the analyzed data and was not able to identify many zero-day attacks. Therefore, these results call for development of more sophisticated clustering techniques, which would address the specifics of data generated in the IoT context.

7.1 Future Work

More research can be done with advanced algorithms that are more appropriate to hefty and involved data from IoT. For example, HDBSCAN and OPTICS are similar to DBSCAN, with more flexible adaptation to different cluster shapes. Spectral clustering and kernel based K means helps in capturing non linear relationship which is more effective for interpreting unusual or rare pattern, which is often indicative of a zero day attack. However, one can also explore hierarchical clustering or fuzzy C means when a data set is high dimensional but would like to see less rigid groupings. In other scenarios, clusters may be further refined with evolutionary or swarm based techniques (e.g. genetic, particle swarm) to further improve accuracy. Moreover, the combination of these clustering methods with supervised learning could bridge the strengths of two approaches and foster the more robust detection of novel threats.

More sophisticated techniques can better capture the hidden relationships in network data than did this study's use of standard methods of feature reduction, such as PCA. As an example, autoencoders can learn a low dimensional (compact) representation of high dimensional traffic pattern and manifold learning methods such as t-SNE or UMAP can uncover non-linear structure in the embedding space. Graph-based embeddings (e.g., node2vec, graph neural networks) may however focus on the more entangled relationships, which are not as amenable to seeing with simpler techniques. Additionally, richer feature representations such as word embeddings (for textual or log based data), as well as other methods like transformer-based encodings, can also be used to improve your anomaly detection outcome.

Currently, the state of the art in clustering, (DBSCAN and K-Means, for instance), don't properly handle how network behavior changes with time. The one that can incorporate changes across various time windows could be an emerging methods like ST-DBSCAN (spatio temporal DBSCAN) or even time aware clustering algorithms. Here, LSTM autoencoders or Temporal Convolutional Networks (TCNs), for example might be used to better track evolving attack vectors for deep learning. These techniques evaluate when, and in what form, particular traffic features arise, simplifying detection of new or uncommon threats that tend to be unanticipated in their evolution (zero-day attacks) among others.

The CICIDS2017 dataset, while comprehensive, represents a single network environment. Future studies could validate the models on multiple datasets or real-world traffic to evaluate their generalisability across different network conditions and attack profiles.

8 References

- Abd, N.S. and Karoui, K., 2024. The importance of the clustering model to detect new types of intrusion in data traffic. *arXiv preprint arXiv:2411.14550*.
- Abdulganiyu, O.H., Tchakoucht, T.A. and Saheed, Y.K., 2024. Towards an efficient model for network intrusion detection system (IDS): systematic literature review. *Wireless Networks*, 30(1), pp.453-482.
- Alfrhan, A.A., Alhusain, R.H. and Khan, R.U., 2020, September. SMOTE: Class imbalance problem in intrusion detection system. In *2020 International Conference on Computing and Information Technology (ICCIT-1441)* (pp. 1-5). IEEE.
- Anthony, R.T., 2023. *Barriers to Adoption of Advanced Cybersecurity Tools in Organizations*. Capitol Technology University.
- Catillo, M., Del Vecchio, A., Pecchia, A. and Villano, U., 2023, August. A case study with CICIDS2017 on the robustness of machine learning against adversarial attacks in intrusion

detection. In *Proceedings of the 18th international conference on availability, reliability and security* (pp. 1-8).

Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P., 2002. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16, pp.321-357.

Cost of a data breach 2024 | IBM. (2024). <https://www.ibm.com/reports/data-breach>

Deng, D., 2020, November. Research on anomaly detection method based on DBSCAN clustering algorithm. In *2020 5th International Conference on Information Science, Computer Technology and Transportation (ISCTT)* (pp. 439-442). IEEE.

Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*.

Fu, Y., Du, Y., Cao, Z., Li, Q. and Xiang, W., 2022. A deep learning model for network intrusion detection with imbalanced data. *Electronics*, 11(6), p.898.

Goodfellow, I., Bengio, Y. and Courville, A., 2016. *Deep Learning*. MIT Press.

Groeger, M. and Waldehagen Berg, L., 2024. Workplace Evolution: The Hybrid Work Model and its Impact on Innovation and Employee Well-Being.

Hastie, T., Tibshirani, R. and Friedman, J., 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. Springer.

Hochreiter, S. and Schmidhuber, J., 1997. Long Short-Term Memory. *Neural Computation*, 9(8), pp.1735-1780.

Jolliffe, I.T. and Cadima, J., 2016. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065), p.20150202.

Jose, J. and Jose, D.V., 2023. Deep learning algorithms for intrusion detection systems in internet of things using CIC-IDS 2017 dataset. *International Journal of Electrical and Computer Engineering (IJECE)*, 13(1), pp.1134-1141.

Khan, Z.I., Afzal, M.M. and Shamsi, K.N., 2024. A comprehensive study on CIC-IDS2017 dataset for intrusion detection systems. *International Research Journal on Advanced Engineering Hub (IRJAEH)*, 2(02), pp.254-260.

Kheddar, H., 2024. Transformers and large language models for efficient intrusion detection systems: A comprehensive survey. *arXiv preprint arXiv:2408.07583*.

Laghrissi, F., Douzi, S., Douzi, K. and Hssina, B., 2021a. IDS-attention: an efficient algorithm for intrusion detection systems using attention mechanism. *Journal of Big Data*, 8(1), p.149.

Laghrissi, F., Douzi, S., Douzi, K. and Hssina, B., 2021b. Intrusion detection systems using long short-term memory (LSTM). *Journal of Big Data*, 8(1), p.65.

Little, R.J. and Rubin, D.B., 2002. *Statistical Analysis with Missing Data*. 2nd ed. Wiley.

Lloyd, S., 1982. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2), pp.129-137.

- Mallick, M.A.I. and Nath, R., 2024. Navigating the Cyber security Landscape: A Comprehensive Review of Cyber-Attacks, Emerging Trends, and Recent Developments. *World Scientific News*, 190(1), pp.1-69.
- Maseer, Z.K., Yusof, R., Bahaman, N., Mostafa, S.A. and Foozy, C.F.M., 2021. Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset. *IEEE access*, 9, pp.22351-22370.
- Mukherjee, S., Neogi, S. and Verma, A., 2024. Adapting to Remote Work: Navigating Sustainability Impacts through Basic Sciences in Indian IT Enterprises. In *Unleashing the Power of Basic Science in Business* (pp. 311-331). IGI Global.
- Muneer, S., Farooq, U., Athar, A., Ahsan Raza, M., Ghazal, T.M. and Sakib, S., 2024. A Critical Review of Artificial Intelligence Based Approaches in Intrusion Detection: A Comprehensive Analysis. *Journal of Engineering*, 2024(1), p.3909173.
- Mustafa, D.H. and Husien, I.M., 2023. Adaptive DBSCAN with Grey Wolf Optimizer for Botnet Detection. *International Journal of Intelligent Engineering & Systems*, 16(4).
- Rahali, A. and Akhloufi, M.A., 2021. Malbert: Using transformers for cybersecurity and malicious software detection. *arXiv preprint arXiv:2103.03806*.
- Ranade, P., Piplai, A., Joshi, A. and Finin, T., 2021, December. Cybert: Contextualized embeddings for the cybersecurity domain. In *2021 IEEE International Conference on Big Data (Big Data)* (pp. 3334-3342). IEEE.
- Rashid, U., Saleem, M.F., Rasool, S., Abdullah, A., Mustafa, H. and Iqbal, A., 2024. Anomaly Detection using Clustering (K-Means with DBSCAN) and SMO. *Journal of Computing & Biomedical Informatics*, 7(02).
- Tahmasebi, M., 2024. Beyond defense: Proactive approaches to disaster recovery and threat intelligence in modern enterprises. *Journal of Information Security*, 15(2), pp.106-133.
- Uddin, R., Kumar, S.A. and Chamola, V., 2024. Denial of service attacks in edge computing layers: Taxonomy, vulnerabilities, threats and solutions. *Ad Hoc Networks*, 152, p.103322.
- Ullah, I. and Mahmoud, Q.H., 2021. A review of intrusion detection systems in IoT: Challenges, approaches and open research issues. *Future Generation Computer Systems*, 122, pp.82-103.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention Is All You Need. *arXiv preprint arXiv:1706.03762*.
- Widodo, A.O., Setiawan, B. and Indraswari, R., 2024. Machine Learning-Based Intrusion Detection on Multi-Class Imbalanced Dataset Using SMOTE. *Procedia Computer Science*, 234, pp.578-583.
- Yang, S., Tan, M., Xia, S. and Liu, F., 2020, June. A method of intrusion detection based on Attention-LSTM neural network. In *Proceedings of the 2020 5th International Conference on Machine Learning Technologies* (pp. 46-50).