International Journal of Artificial Intelligence
p-ISSN: 2407-7275, e-ISSN: 2686-3251

Original Research Paper

Network Anomaly Detection System using Transformer Neural Networks and Clustering Techniques

Ayomitope Isijola[1*], Michael Asefon[2], Ufuoma Ogude[1], Adetoro Mayowa Sola[3], Temiloluwa Adebowale[1], Isabella Akunekwu[4]

[1] Department of Computer Sciences, University of Lagos. Lagos, Nigeria.
[2] Department of Computer Science, National Open University of Nigeria. Nigeria.
[3] Department of Electrical/Electronics and Computer Engineering, College of Engineering Afe Babalola University Ado. Ekiti State, Nigeria.
[4] Department of Information Management Technology, Federal University of Technology Owerri. Imo, Nigeria.

Article History
Received:
13.05.2025

Revised:
01.06.2025

Accepted:
23.06.2025

***Corresponding Author:**
Ayomitope Isijola
Email:
ayomitopeisijola@yahoo.com

Abstract: This study proposes a hybrid approach for network anomaly detection by integrating a Transformer-based model with clustering techniques. The methodology begins with the application of K-means clustering as a preprocessing step to group similar network traffic data, thereby reducing data complexity and highlighting significant patterns. The clustered data is then fed into a Transformer model, which utilizes multi-head self-attention mechanisms to capture intricate temporal dependencies and contextual relationships within sequential data. This dual-stage approach enhances the model's ability to differentiate between normal and anomalous behaviors in network traffic. Trained on a network security dataset, the system effectively identifies both common and rare attack types. According to the results, the suggested ensemble classifier outperformed existing deep learning models with an accuracy of over 99.5%, 98.5%, and 99.9% on the UNSW-NB15 dataset. The synergy between the unsupervised pattern recognition of clustering and the deep learning capabilities of Transformers enables a scalable and adaptable solution for real-world network security applications, making it suitable for proactive cyber threat detection and mitigation.

Keywords: Deep Learning, K-means Clustering, Network Anomaly Detection, Neural Network Architecture, Transformer Model.

1. Introduction

Traditional network security methods such as antivirus software, firewalls, spyware detection, and authentication mechanisms offer protection across multiple layers of a network, yet intrusion attacks remain a persistent threat. However, with the constant evolution of technology, network security has gained more attention, emphasizing the importance of safeguarding digital life and data. Intrusion detection remains a crucial component of network security, with traditional systems relying on anomaly detection that analyzes unusual traffic patterns to identify potential breaches. Machine learning-based methods have become a significant aspect of contemporary intrusion detection, offering more accurate detection of uncommon activity and possible intrusions by studying normal network traffic styles [1]. Intrusion detection systems (IDS) play a vital role as they provide an effective mechanism to avert or put a stop to cyberattacks. Recent developments in artificial intelligence (AI) have led to numerous deep learning techniques for intrusion anomaly detection aimed at enhancing network security. Lv & Ding [2] presented a novel hybrid framework called KCLSTM, merging the K-means clustering algorithm with a convolutional neural network (CNN) and long short-term memory (LSTM) architecture for the binary classification of intrusion detection systems.

Zhu et al. [3] proposed a novel hybrid deep learning model, FC-Trans, created to improve network intrusion monitoring. Their technique involves optimizing feature representation using the Feature Tokenizer method, leveraging CNNs to take out meaningful features from the data, and embodying the Transformer’s self-attentive mechanism and residual structure to apprehend long-term feature dependencies and alleviate gradient vanishing. One common machine learning approach for intrusion detection involves anomaly detection systems (ADS), which identify inappropriate, inaccurate, or anomalous activities in network environments by looking at a variety of data records seen in processes on the system. Most recent anomaly detection techniques are based on unsupervised learning due to recent developments in unforeseen anomalies with generative adversarial networks. However, the aforementioned methods are obsolete in detecting aberrant information since they can only represent local information due to the convolution operation's narrow receptive field, which also limits their capacity to catch long-range details in the data.

A transformer's attention module can link the input sequence to study long-term information. This capability allows it to measure the significance of each part of the input sequence, permitting the model to understand connections that span across distant elements. As a result, Transformers are well-suited for detecting abnormal information that is distributed both locally and globally, making them more versatile and powerful compared to traditional models with limited receptive fields, like convolutional or recurrent networks [4].

Network Anomaly Detection concentrates on determining rare or unexpected patterns that diverge from normal behavior in the system traffic, the unexpected pattern could be because of a network breach or a malicious attempt on the network. The research aims to improve a network security anomaly detection system that utilizes a Transformer-based neural network with the use of clustering techniques. The model with clustering techniques enhances anomaly detection by providing better data structuring, improving detection accuracy, reducing computational complexity, and offering greater interpretability. Clustering helps the transformer model operate on a cleaner, more manageable version of the dataset, allowing it to focus on meaningful patterns and improving its overall efficiency.

The study aims to evaluate the performance of a transformer-based neural network integrated with clustering techniques, comparing it to a standard transformer model without clustering. It highlights the efficacy of anomaly detection within network environments. Its objectives include:

(1) to improve on network anomaly
(2) to compare, contrast, and begin utilizing results, the instrumental precision that the combined modeling approach with transformer-based methods achieves in identifying threats in comparison to single classifiers.

This research focuses on implementing a practical application of a Transformer-based anomaly detection system as an Intrusion Detection System (IDS). These datasets—UNSW-NB15, NSL-KDD, and CSE-CIC-IDS2018—are considered for this purpose. The approach is a two-step procedure: In the first stage, we will collect and process these datasets, ensuring they include a comprehensive range of both normal and anomalous network behaviors. The preprocessing will involve cleaning the data, feature extraction, and normalization to prepare it for model training. In the second stage, we will design and implement a model that combines transformer models with neural networks to leverage their strengths in catching long-term dependencies and intricate patterns. The model will be pre-trained on large, unlabeled data to learn general network behavior and fine-tuned on the labeled datasets to enhance its anomaly detection capabilities. We will analyze the model's effectiveness utilizing metrics like accuracy, detection rate, and computational efficiency, comparing it with existing IDS approaches. Finally, we will use attention mechanisms to interpret the model's decisions, providing insights into the detection process to ensure the system is both effective and explainable.

The study scope is to propose an anomaly detection system based on a Transformer Neural Network Architecture, integrating Unsupervised Machine Learning (ML) techniques, specifically clustering algorithms, to mitigate cybersecurity threats. This approach aims to compare the effectiveness of the combined modeling approach with transformer-based techniques against single neural network architectures and traditional unsupervised ML clustering classifiers, establishing instrumental precision in threat identification through empirical results analysis. The primary objective is to improve the detection of both common and rare cyber threats in network traffic. The original contribution lies in combining unsupervised clustering to reduce data complexity and uncover latent patterns, with the deep contextual learning capabilities of Transformers to effectively capture temporal dependencies. This hybrid model achieved superior performance across key metrics—accuracy (>99.5%), precision, recall, F1-score, and AUC-ROC—on the UNSW-NB15 dataset, outperforming baseline and conventional deep learning models. The approach also enhances interpretability, offering a scalable solution for proactive network security in real-world environments.

The limitations of the study are its potential lack of generalizability, as the effectiveness of clustering and transformer-based neural network models in anomaly detection within network environments may vary across different network setups, datasets, and attack scenarios, thereby limiting the applicability of the findings to specific contexts. Additionally, the study may not comprehensively account for all possible variations and nuances within real-world network environments, potentially affecting the interpretation and practical application of its results.

2. Literature Review

In previous years, machine learning techniques have been comprehensively utilized in the network security domain thanks to their efficient automated feature extraction techniques. Researchers have utilized machine learning and AI algorithms to detect network intrusion.

The network anomaly identification process aims to determine when network behavior deviates from normal habits. The identification of unusual events in vast dynamic networks has grown in significance as networks get bigger and more intricate. Nevertheless, it is quite hard to identify network anomalies swiftly and precisely.

2.1. Deep Learning Techniques

Deep learning's strong feature modeling capabilities make it a promising technique for network anomaly identification. By leveraging the strengths of CNNs, BLSTMs, and Transformers, along with clustering techniques, you can develop a robust and accurate network anomaly detection system that is capable of identifying a wide range of malicious activities and unusual behaviors in network traffic.

Deep learning has shown greater efficacy in recognizing network anomalies. Notably, a recurrent neural network model was designed to identify serial data patterns for prediction. The hybrid model was optimized, and the convolutional neural network merged with Bidirectional Long-Short Term Memory (BLSTM), to study optimizers (Adam, Nadam, Adamax, RMSprop, SGD, Adagrad, Ftrl), number of epochs, size of the batch, learning rate, and the Neural Network (NN) architecture. Inspecting these hyperparameters produced the highest accuracy in anomaly detection, reaching 98.27% for the binary class NSL-KDD and 99.87% for the binary class UNSW-NB15 [5].

Zhou et al. [6] presented a network traffic anomaly detection model contingent on feature grouping and multiple autoencoders (multi - AEs) integration. This model consists of four modules: feature grouping module, feature learning module, Area Under the Curve (AUC) and optimal threshold calculation module, and anomaly detection application module. In the feature grouping module, multiple group features are built by choosing different features according to their attributes and variances. In the feature learning module, the group features of normal traffic are learned based on multi - AEs. In the AUC and optimal threshold calculation module, the AUC of each AE was calculated according to the ROC curve of the verification data, and the optimal thresholds for each AE were determined by utilizing the Youden index. In the anomaly detection application module, the AEs that engaged in fusion are chosen, and their weights are derived by examining the AUC value, and the scores of undisclosed traffic in each AE are accessed considering both the reconstruction error distribution and the optimal threshold. Finally, the anomaly detection result can be derived by the fusion of these multiple scores. Through validation on the UNSW - NB15 and CICIDS2017 datasets, the accuracy of the proposed model was boosted by 12.04% and 10.52%, respectively, compared to the baseline model.

The dispersed architecture of cloud computing entails strong defense mechanisms to protect network-accessible resources against a diverse and dynamic threat landscape. A Network Intrusion Detection System (NIDS) is crucial in this context, with its effectiveness in cloud environments depending on its adjustability to evolving threat vectors while alleviating false positives. Hence, Long et al. [7] presented a novel NIDS algorithm, berthed in the Transformer model and finely tailored for cloud environments. The algorithm blends the basic aspects of network intrusion detection with the advanced attention mechanism essential to the model, easing a more insightful inspection of the relationships between input features and diverse intrusion types, thereby strengthening detection accuracy.

Tuli et al. [8] introduced a Transformer based neural network model (TranAD), a deep learning model depicted for effective identification and recognition of anomalies in multivariate time series data, focusing on key problems like lack of labeled anomalies, high data volatility, and the need for low-latency inference. TranAD utilizes a Transformer-based architecture with attention-based sequence encoders to capture broader temporal patterns and perform quick inference. It employs self-conditioning for vigorous feature extraction and adversarial training for consistency. Also, Model-Agnostic Meta-Learning (MAML) permits the model to be trained with finite data.

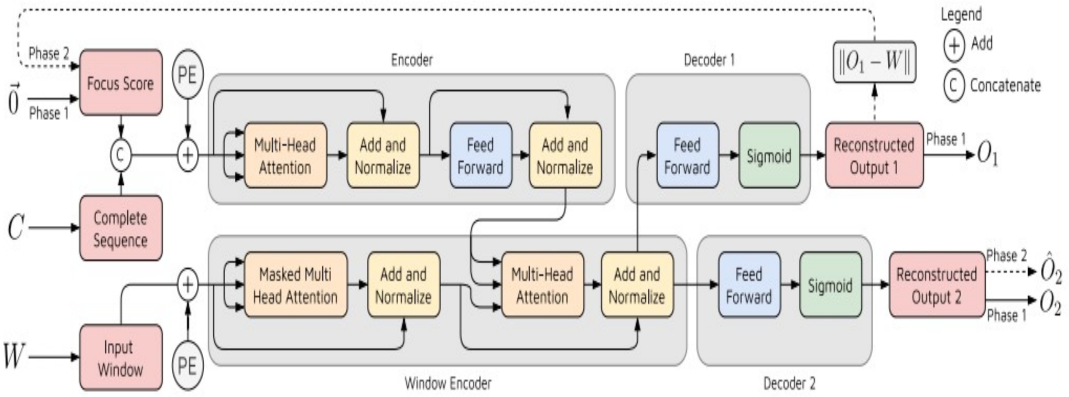

Figure 1. The Architecture of a Transformer-Based Model Deep Neural Network for Anomaly Detection [8]

2.2. Traditional Machine Learning Techniques

Understanding the performance and limitations of traditional machine learning algorithms, such as K-Nearest Neighbors (KNN), Support Vector Machines (SVM), and Naïve Bayes, is essential for designing and evaluating our proposed network anomaly detection system. This understanding enables us to compare our approach with existing methods, justify its use, and potentially create hybrid solutions that combine the strengths of both traditional and deep learning techniques.

Saran & Kesswani [1] conducted a study comparing various supervised Machine Learning classifiers for Intrusion Detection. The rapid expansion of computer networks and the internet has led to heightened security concerns, despite traditional defenses like antivirus software and firewalls. Previous research has proposed Intrusion Detection Systems (IDS) using ML classifiers, yet they have often encountered challenges such as outdated datasets and limited coverage of attack types. In response, the study introduced an IDS employing multiple ML classifiers on the MQTT-IoT-IDS2020 dataset. Experimental results demonstrated high accuracy rates ranging from 97.58% to 99.98% across classifiers like K-Nearest Neighbor, Support Vector Machine, Naive Bayes, Random Forest, Decision Tree, and Stochastic Gradient Descent. The study also assessed Precision, Recall, and F1-Score metrics, highlighting the IDS's efficacy in detecting a variety of intrusion attacks in IoT environments. The researchers conducted extensive empirical studies using six publicly available datasets to evaluate the effectiveness of TranAD. Their discovery showed that TranAD outperforms current state-of-the-art baseline techniques in both identification and diagnosis assignments. Additionally, TranAD is more effective in terms of data and training time. Specifically, the research revealed that TranAD can improve F1 scores by as much as 17% and reduce training durations by as much as 99% in comparison to the baseline methods.

2.3. Hybrid and Metaheuristic Optimization Models

These hybrid models are important as they demonstrate the advantages of combining clustering techniques with Transformer architectures to enhance anomaly detection systems. By integrating K-means clustering, our system effectively reduces data complexity and highlights significant patterns, which improves the Transformer model's ability to differentiate between normal and anomalous network behaviors. Additionally, incorporating Generative Adversarial Networks (GANs) with Deep Reinforcement Learning (DRL) can help address data imbalance challenges and improve the detection of rare attack types. GA, GWO, and GW-GA enhance the feature quality of our network anomaly detection system before clustering and also fine-tune the transformer architecture, ultimately increasing the system's robustness and accuracy.

Al-Dahoul et al. [9] addressed the persistent challenge of network anomaly detection using a large-scale and highly imbalanced dataset. Their approach involves training deep neural networks with improved class weights to effectively study complex patterns from scarce anomalies in noisy and imbalanced network traffic data.

They introduce a new model fusion technique that blends two deep neural networks: one for binary classification of normal versus attack traffic, and another for multi-class classification of various attack types. The utilized method successfully identifies a range of network attacks, including Distributed Denial of Service (DDoS), IP probing, PORT probing, and Network Mapper (NMAP) probing. The experimental results on the ZYELL dataset unveil that the model achieved better performance than baseline models. Their approach improved the average macro Fβ score by 17%, indicating higher accuracy in detecting network anomalies across different types.

Additionally, they reduced the false alarm rate by 5.3%, which means fewer instances of incorrectly flagging normal network traffic as anomalous.

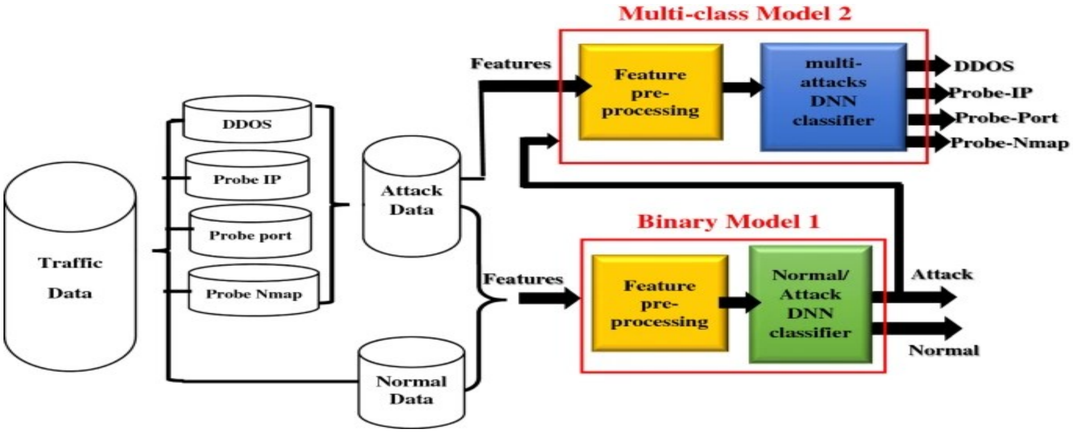

Figure 2. The Model for a Network Anomaly Merges Two Deep Neural Networks:
A Binary Normal/Attack Classifier and A Multi-Attack Classifier.
Model Fusion of Deep Neural Networks for Anomaly Detection [9]

Benaddi et al. [10] presented a novel approach aimed at improving the effectiveness and efficiency of Intrusion Detection Systems (IDS) within Industrial Internet of Things (IIoT) networks. Their method integrated Distributional Reinforcement Learning (DRL) and Generative Adversarial Network (GAN) techniques to bolster IDS capabilities, particularly in detecting minority attacks and addressing data imbalance challenges. Evaluation using real-world datasets validated the exceptional performance of the DRL-GAN models in contrast to common DRL approaches. Significant improvements were observed through key measures like accuracy, precision, recall, and F1 score for both binary and multiclass anomaly detection classifications. This underscored the possibility of leveraging advanced machine learning techniques like DRL and GANs to strengthen cybersecurity defenses in IIoT environments against evolving cyber threats.

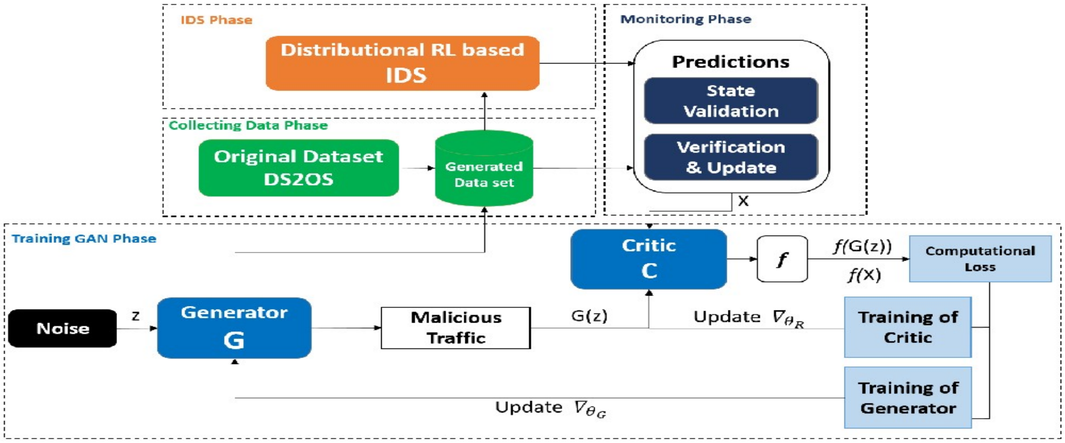

Figure 3. The Flow Diagram for DRL and GAN Techniques in Identifying Minority Attacks on IoT Devices, Anomaly Detection in Industrial IoT by Utilizing Distributional Reinforcement Learning and Generative Adversarial Networks [10]

Kunhare et al. [11] conducted a study on an Intrusion Detection System (IDS) that integrates hybrid classifiers with meta-heuristic algorithms for optimization and feature selection, specifically employing genetic algorithms. The research underscores the critical role of IDS in identifying threats and unauthorized access through network traffic monitoring. Inefficient features within IDS systems can significantly impede operational efficiency and delay accurate decision-making processes. The study explores various approaches, including machine learning algorithms, data mining, swarm intelligence, and artificial neural networks, aimed at enhancing IDS capabilities. A new feature selection technique utilizing genetic algorithms was proposed, alongside the implementation of hybrid classification techniques combining logistic regression and decision trees. This approach aims to enhance identification rates and overall validity in intrusion detection. Comparative analysis of different meta-heuristic algorithms highlights the efficacy of the grey wolf optimization algorithm, demonstrating superior performance in achieving high accuracy and detection rates while reducing the number of features required. The study focuses on the grey wolf optimization algorithm among other meta-heuristic approaches. The performance and applicability of various algorithms, like particle swarm optimization or ant colony optimization, in various network intrusion scenarios, were not extensively compared or evaluated.

With the rise in communication in an IoT system, internet security has lessened, and the most dangerous and advanced attacks in the IoT have come into view, i.e., DDoS and Botnet attacks. DDoS attacks are a significant threat to the accessibility of internet services, particularly since botnets can now be initiated by almost anyone. In this situation, the use of an intrusion detection system (IDS) is crucial to identify intruders and maintain the security of IoT networks. Maazalahi & Hosseini [12] proposed a new IDS to identify IoT-Botnet DDoS attacks. This IDS is a new three-phase system, the first phase is related to preprocessing on the dataset, and the second phase involves a new hybrid method for feature selection by utilizing filter and wrapper methods based on the Grey Wolf (GW) algorithm and genetics called GW-GA. In this technique, the initial population is randomly selected, and then at each stage, feature selection is done by both algorithms at the same time the final answer was compared and the best results are given as a new population to both algorithms and the third phase covers the use of machine learning and metaheuristic algorithms as classifiers.

Davahli et al. [13] presented a lightweight machine learning-based intrusion detection method with high efficiency for resource-limited IoT wireless networks, i.e., IoT intrusion detection system (IoTIDS). IoTIDS is contingent on the hybridization of a genetic algorithm (GA) and grey wolf optimizer (GWO), termed GA–GWO. The main objective of the hybrid algorithm for the IoTIDS is to decrease the dimensionality of the huge wireless network traffic through the brilliant selection of the most explanatory traffic features. By hybridizing, we try to remove their weaknesses through GA and GWO strengths. The efficacy of the GA–GWO on IoTIDS was examined using AWID (Aegean wi-fi intrusion dataset) as a new real-world wireless intrusion dataset, after preprocessing it under different schemes. The experimental results show that the proposed GA–GWO individually not only enhanced the efficacy of the IoTIDS in terms of computational costs but also allowed the IoTIDS to identify with high accuracy and low false alarm rate.

2.4. Domain-Specific Applications

Space communication networks and MQTT-based IoT systems face high vulnerability, complex architectures, and limited labeled data, making accurate and real-time anomaly detection essential. Our hybrid model, which combines Transformers and K-means clustering, offers a scalable, adaptive, and efficient solution by learning complex patterns, processing large data volumes quickly, and reducing reliance on labeled datasets.

Diro et al. [14] highlighted the crucial role of anomaly detection in protecting communication and networking systems, particularly in space environments. The research examines the growing cyber threats to space systems and identifies significant challenges such as scalability and real-time detection. By reviewing current anomaly detection techniques, the study aims to enhance detection accuracy and resilience against threats. A noteworthy contribution is the proposed hybrid approach that combines stream-based and graph-based techniques to tackle the complexity of space communication networks. The survey ultimately provides valuable recommendations for improving anomaly detection mechanisms in space-based communication and networking systems.

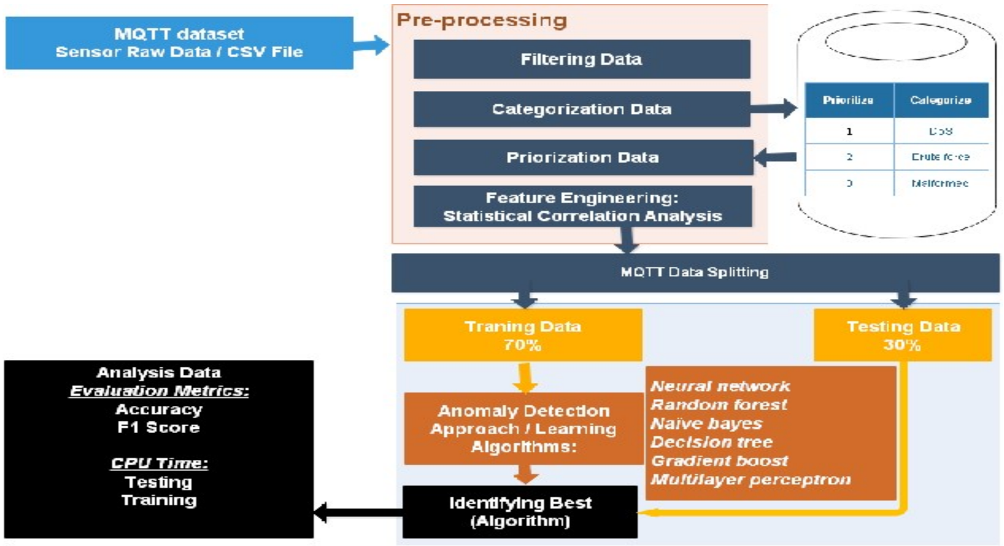

Figure 4. A Diagram Depicting the Dialog Flow of The MQTT Protocol, Improving Accuracy for Identifying Anomalies in the MQTT Network by Applying Correlation Analysis for Feature Selection using Machine Learning Techniques [15]

Imran Ali et al. [15] conducted research addressing the growing significance of anomaly detection (AD) within Internet of Things (IoT) applications, with a specific concentration on the Message Queuing Telemetry Transport (MQTT) protocol. The study underscores the critical necessity for the timely detection of assaults like brute-force, Denial-of-Service (DoS), malformed, flood, and SlowITe assaults to preempt irreversible damage. It explores methods to expedite attack prediction, proposing an innovative method utilizing correlation analysis to streamline training and testing times for machine learning algorithms. Evaluation across various algorithms highlights substantial gains in feature engineering, particularly in identifying pertinent features within MQTT datasets. Notably, this research marks a pioneering effort in reducing prediction times for DoS and malformed attacks within MQTT networks, underscoring the imperative of efficient anomaly detection for safeguarding IoT environments.

Existing research typically focuses on either:

1) Traditional Machine Learning Approaches: Methods like k-means clustering, PCA, and rule-based systems were used for anomaly detection, but often struggle with complex sequential patterns in network data.
2) Deep Learning Models: Techniques like LSTMs and CNNs have been used for detecting anomalies, but they may not fully capture long-range dependencies and relationships in network traffic.
3) Supervised and Unsupervised Learning: Most current approaches rely on supervised learning, requiring labeled datasets, which limits their ability to detect zero-day attacks.
4) This study bridges these gaps by combining k-means clustering (unsupervised feature preprocessing) with transformers (deep learning for sequential patterns). The novelty lies in leveraging multi-head self-attention to enhance anomaly detection accuracy, particularly for detecting previously unseen attack types. This hybrid approach makes the model more adaptable and scalable for real-world cybersecurity applications.

3. Methodology

This study aims to improve anomaly detection by leveraging the strengths of both clustering techniques and transformer models. Specifically, this study combines the inherent capabilities of clustering, such as simplifying data structure, reducing noise, and grouping similar data points, with powerful sequence modeling and attention.

Mechanisms of the transformer-based neural network. By integrating these two approaches, the study aims to overcome common challenges in anomaly detection, such as the high-dimensional nature of data, the presence of imbalanced datasets, and the difficulty of accurately distinguishing between normal and anomalous patterns.

The study is structured into four stages, which are data preparation, clustering with K-Means, anomaly identification utilizing a pre-trained Transformer model, and data visualization.

The K-Means algorithm was utilized for clustering the data points, followed by anomaly identification, making use of the pre-trained Transformer model algorithm. By combining the strengths of both algorithms, this study aims to provide improved anomaly detection in Network Intrusion Detection Systems.

3.1. Data Collection

In this research, we utilize an open-sourced dataset, namely the UNSW-NB15, NSL-KDD, and CSE-CIC-IDS2018. This dataset is extensively utilized in the field of network security and provides a rich collection of network traffic data for analysis and evaluation purposes.

The data extraction process involves parsing and extracting packet-level information from PCAP (Packet Capture) files. PCAP files store network traffic data captured during network monitoring or analysis. By incorporating the UNSW-NB15, NSL-KDD, and CSE-CIC-IDS2018 datasets, the research benefits from the diverse and realistic nature of the included network traffic data.

3.2. Data Preprocessing

The data preprocessing phase is essential to prepare the dataset for effective model training and assessment. This action includes label encoding, normalization, and batch processing. The dataset is first split into a 7:3 ratio, ensuring that 70% of the data is used for training and 30% for testing. This ensures a well-balanced distribution for both training and evaluating the model. Below are the steps involved:

- Label Encoding: Labels in the raw dataset are strings, which must be converted into an analytical representation for machine learning models. Label encoding transforms these labels into one-hot vectors, where each group stands for a binary vector. This conversion enables efficient data processing by the model.
- Normalization: To ensure consistent scaling of the dataset features, normalization was applied. This step addresses the issue of differing value ranges in the dataset, which could negatively impact model training. We use the Min-Max Scaler function from sklearn to rescale the values in each feature to a range of 0 to 1. This preserves the relationships in the data but guarantees that each feature contributes proportionately to the learning process of the model.

  The normalization is calculated as follows:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \qquad (1)$$

  where:
  x is the value of a data point,
  x min and x max stand for the minimum and maximum values of the column,
  x scaled is the normalized value
  This ensures that all features are on a comparable scale, making it simpler for the model to learn efficiently.

- Batch Processing: Given the large size of the dataset, it is impractical to feed all data points into the model at once due to memory constraints and potential slowdowns. Batch processing was employed to split the dataset into smaller, manageable chunks, allowing the model to process the data in stages. The dataloader function from the PyTorch library is utilized for this purpose.

  The process is broken as follows:
  1) Convert Data: The training data is first transformed into a format recognized by PyTorch using the Tensor Dataset function. The converted dataset is then passed into the dataloader function, which generates an iterator that can handle batch-wise training. The batch size is set via the batch size parameter, which determines how many samples are passed into the model at once.
  2) Batch-wise Data Feeding: Finally, the iterator was used to fetch smaller batches of data for training, ensuring that the model can train without running into memory issues while improving overall efficiency.

3.3. K-Means

K-means ranks among the most extensively used unsupervised machine learning algorithms, particularly for clustering tasks. It works by dividing a dataset into K different clusters, where every data point is a member of the cluster most closely related to the centroid.

The algorithm seeks to reduce the length between clusters while optimizing the length between data points inside the same cluster. It helps to reduce noise and organize the data by putting related data points together, which is especially useful in the pre-processing phase before more complex models are applied. Key components of k-means clustering:

- Centroid Initialization: The K-Means algorithm initializes K centroids randomly, which serve as the starting points for clusters. Proper initialization is crucial for optimal clustering. The centroids are iteratively updated to refine cluster assignments. The number of clusters (K) is a hyperparameter chosen based on the dataset and application, such as distinguishing normal and intrusive network traffic in intrusion detection.
- Distance Calculation: Once the centroids are set, the algorithm calculates the separation of each data point in the dataset and each centroid. The most often used measure of distance is

Euclidean distance. Each data point is then assigned to the nearest centroid, forming a cluster.

- Distance Metric: The distance metric defines how the similarity between a data point and the centroid is measured. Euclidean distance is frequently used, though other measures such as Manhattan distance can also be applied.

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(pi - qi)2} \tag{2}$$

where:
p and q are two points in the dataset.
n is the dimensionality of the data.

- Cluster Assignment: Data points are allocated to the closest centroid, creating K clusters of similar points. Intra-cluster similarity ensures that points within a cluster are alike, while inter-cluster separation distinguishes different clusters. This enhances the differentiation of normal and anomalous traffic in anomaly detection.
- Centroid Update: Once all data points have been allocated to clusters, the centroids of each cluster are reevaluated. The new centroid is the middle of all the points allocated to that cluster. This process is crucial as it helps K-Means adjust the cluster centers to better fit the data.
- Iteration and Convergence: The K-Means algorithm iterates through distance calculations, cluster assignments, and centroid updates until convergence, where centroids stabilize. In network traffic analysis, the clusters formed may represent different types of activity, including normal traffic and suspicious behavior like network intrusions or Distributed Denial of Service (DDoS) assaults. Once the clusters are stable, the data is passed on to the Transformer model for further anomaly detection. The clustering process reduces noise in the data and helps the Transformer focus on specific types of traffic behaviors, improving its ability to detect anomalies. This study employs the K-means clustering technique because of its simplicity, computational effectiveness, and capacity to scale with large, high-dimensional datasets. K-means effectively groups network traffic into meaningful clusters, distinguishing between normal activity and abnormal or intrusive behaviors. K-Means was used as a preprocessing step to group data before passing it to the Transformer model for anomaly detection. The Transformer excels at analyzing sequential data and detecting complex patterns through its attention mechanism, which helps identify normal and abnormal network behaviors effectively.

3.4. Transformer-based Neural Network

The Transformer consists of several key components that contribute to its effectiveness in anomaly detection:

- Input Embedding: Clustered network traffic data is transformed into embeddings through input embedding, mapping numerical values into a higher-dimensional vector space. This process helps the Transformer model record meaningful connections between data points for effective analysis.
- Positional Encoding: Transformers are not inherently aware of sequence, so positional encodings are attached to embeddings in network traffic analysis. This preserves the order of packets or connections, allowing the model to record temporal dependencies crucial for detecting anomalies over time.
- Self-Attention Mechanism: The primary invention of the Transformer is its self-attention mechanism. Self-attention permits the model to concentrate on distinct parts of the input data to identify which parts are most relevant to the task.
  For anomaly detection, this means the Transformer can look at a sequence of network events (e.g., packet flows) and determine which specific events (or clusters of events) are most indicative of normal behavior and which might signify an anomaly. The self-attention process works by:
  1) Querying: Each data point (clustered traffic) is compared to every other point to calculate relevance.
  2) Keying: It identifies which other points in the data sequence might provide important

context for the current point.

3) Valuing: It then assigns attention weights to different points based on their relevance and uses these to focus on important patterns. This allows the Transformer to catch both temporary and permanent dependencies in network traffic, improving its ability to spot deviations that suggest anomalies.
4) Multi-Head Attention: The multi-head attention mechanism increases self-attention by allowing the model to concentrate on several facets of network traffic simultaneously. Different heads analyze temporal correlations, packet sizes, or sources, ensuring a comprehensive detection of potential anomalies.
5) Feedforward Neural Network: After the attention layers, the data passes through a standard feedforward neural network. This unit processes the weighted outputs from the attention layers and combines the information to make predictions. The network is responsible for learning complex patterns in the data that might indicate an anomaly.
6) Layer Normalization and Residual Connections: Transformers involve layer normalization and residual connections between layers to balance and accelerate the training process. These techniques help prevent the model from overfitting on large datasets, ensuring that it generalizes well to unseen data, which is crucial for anomaly detection.

3.5. Model Training

Upon receiving the clustered dataset, our proposed Transformer architecture initiates the training process by dividing the dataset into two distinct divisions: a training set and an evaluation (or validation) set. This division follows a 90% to 10% ratio, meaning that 90% of the data will be utilized to train the model, while the 10% that remains will serve to evaluate its performance. Data preprocessing steps include:

- Encoding Categorical Features: The Transformer encodes categorical features using one-hot encoding and scales numerical features using min-max scaling after a logarithmic transformation to reduce skewness. This preprocessing step ensures consistent feature representation and improves model performance by handling categorical data and scaling numerical features to a common range.
- Fit and Transform Methods:
    1) We implement specific fit and transform techniques for both analytical and explicit fields. The fit method learns the parameters (e.g., mean, standard deviation for numerical features; categories for categorical features) from the training dataset, while the transform method applies these learned parameters to preprocess both the training and evaluation datasets.
    2) The expected format for categorical features is provided as a parameter to these methods. This flexibility allows for dynamic switching of preprocessing techniques based on the input encoding, catering to various types of categorical data formats.
- Input Encoder Consideration: The first layer of the Transformer architecture was designed to consider the explicit format as anticipated by the input encoder. This ensures that the model is appropriately configured to handle the specific data types being fed into it.
- Transformer Hyperparameters: In addition to evaluating different transformer topologies and associated hyperparameters, we also compared input encoding strategies and classification heads. These are detailed in Table 1.

Table 1. Transformer Model with Clustering Technique Approach Hyperparameters

Hyperparameters	Values
n_clusters	3; 7; 8
Transformer Block	Encoder, Decoder
Layers	2; 4; 6; 8
Feed Forward (FF) Dimensions	128
Attention Heads	2, 4
Learning Rate	0.01; 0.001; 0.0005

4. Finding and Discussion

This section evaluates the efficiency and feasibility of the proposed anomaly detection model, which integrates clustering techniques with Transformers. A series of experiments was conducted to assess the model's effectiveness, efficiency, and robustness. The process includes setting up the experimental environment, describing the dataset, detailing the data preprocessing methods, and presenting the performance metrics used for evaluating the model. Lastly, the results are compared with alternative approaches to highlight the advantages of the proposed method.

4.1. Experimental Setup

The experiments were performed in a Python environment using the PyTorch deep learning framework, which was chosen due to its powerful GPU support and flexibility in handling complex neural networks. The following hardware and software configurations were used:

- Hardware Configuration:
    1) CPU: Intel Core i7 or higher
    2) RAM: 16GB or more for handling large datasets
- Software Configuration:
    1) Operating System: Windows 10
    2) Python Version: 3.x
    3) TensorFlow
    4) Supporting Libraries: sci-kit-learn, numpy, matplotlib, and pandas for data processing, normalization, and visualization.

4.2. Dataset Description

The proposed model was evaluated using a publicly available anomaly detection dataset containing both normal and anomalous samples. The dataset includes categorical and continuous variables, with the target label indicating anomalies. To ensure a robust evaluation, the data was split into 70% for training and 30% for testing.

4.3. Optimal Number of Clusters

The Elbow Method was used to determine the optimal number of clusters by evaluating the Within-Cluster Sum of Squares (WCSS) across different cluster counts. The "elbow" point in the plot indicated that the optimal number of clusters was three. This value is then used for data clustering as a preprocessing step.

4.4. Preprocessing

Before feeding the dataset into the model, a comprehensive preprocessing pipeline was employed, as discussed earlier. This included label encoding, normalization, and batch processing.

- Label Encoding: The categorical target labels were converted into one-hot encoded vectors, enabling numerical processing and improving computational efficiency.
- Normalization: Feature scaling was performed using the Min-Max Scaler to normalize values between 0 and 1, preventing any feature from dominating the training process.
- Batch Processing: The dataset was processed in small batches using PyTorch's dataloader, improving training speed, reducing memory consumption, and enabling faster convergence.

4.5. Experimental Results

The K-means clustering results reveal distinct groupings among the various attack types in the UNSW-NB15, NSL-KDD, and CSE-CIC-IDS2018. By setting the optimal cluster number (K) to 6, as determined through the Elbow Method, K-means effectively groups data points based on their similarities in attack behavior. Each cluster contains a mixture of attack types, with normal and generic attack types dominating the clusters, reflecting their higher frequency in the dataset. Other attacks, such as DoS, Exploits, and Reconnaissance, are more sparsely distributed among clusters.

The distribution highlights that while some attacks share similar patterns, others exhibit unique characteristics and form distinct groups. The clustering results provide valuable insights into the natural structure of attack behaviors, offering a foundation for further analysis, such as anomaly detection. However, some overlap between clusters suggests that certain attack types share similar features, indicating that K-means alone may have limitations in separating closely related attacks without additional processing steps, such as using a classification model. In summary, centroids provide a condensed view of the dataset's structure, showing the primary characteristics of each cluster and highlighting key features that differentiate types of attacks. This information is valuable both for understanding attack patterns and for further refining clustering or anomaly detection models.

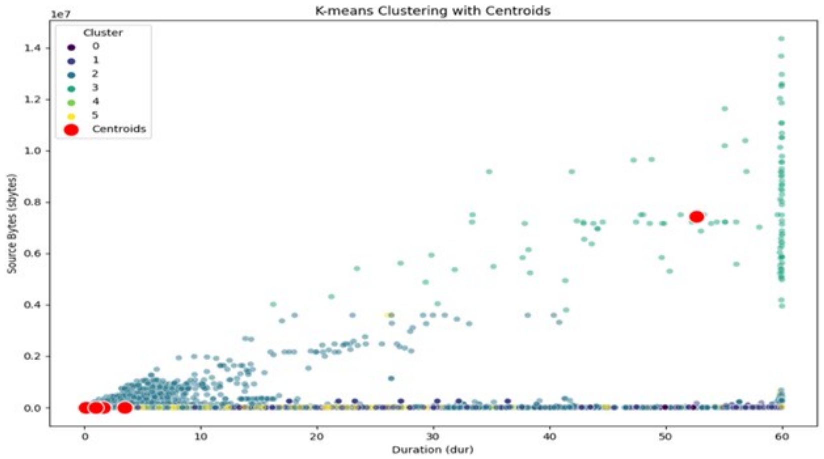

Figure 5. The Scattered Diagram for the K-Means Clustering with Centroids on the Dataset

After clustering with K-means, the Transformer model was used to classify each data point and detect anomalies inside the dataset. By manipulating the attention mechanism within the Transformer, the model identified subtle patterns and dependencies across features, allowing it to distinguish between normal and anomalous instances effectively.

The Transformer model successfully detected anomalous patterns across the different attack types, showing a high level of sensitivity to rare and complex attack behaviors, such as Backdoor and Shellcode attacks. Its attention mechanism helped focus on critical features that are commonly overlooked by simpler models, enhancing detection accuracy.

When used as a classifier, the Transformer demonstrated a strong capability to differentiate between the attack types grouped by the K-means clusters. It assigned accurate labels to instances by capturing nuanced relationships between features, particularly in distinguishing similar types, like Reconnaissance and Exploits, which often share overlapping characteristics.

The model's attention provided insights into which features were most important for detecting specific attack types, allowing for interpretability in understanding why certain instances were classified as anomalies or attack types.

Overall, the Transformer model added depth to the clustering analysis, effectively classifying attacks and identifying anomalies within each cluster, making it a powerful tool for sophisticated threat detection in network security.

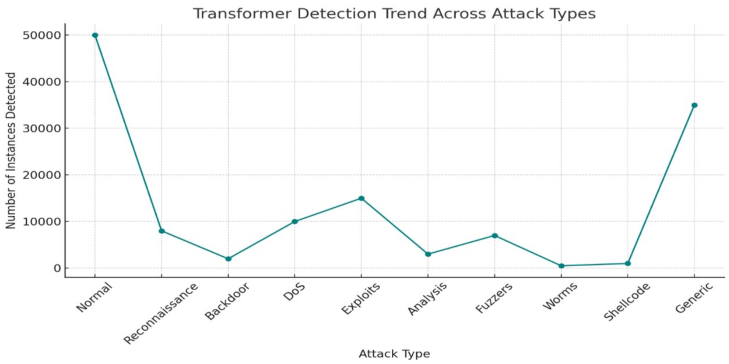

Figure 6. The Line Chart of the Number of Different Attacks a Transformer Model Could Detect on the Dataset

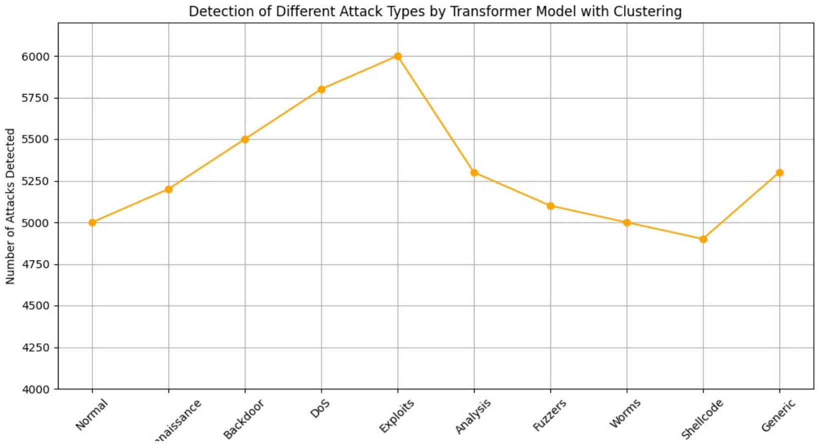

Figure 7. The Line Chart of the Number of Different Attacks a Transformer Model with Clustering Could Detect on the Dataset

4.6. Evaluation Metrics

To measure the model's performance, several widely used metrics were employed:

- Accuracy: The percentage of correctly classified cases, both normal and anomalous, out of all the cases.
- Precision: The percentage of correctly identified anomalies out of all cases labeled as anomalies.
- Recall (Sensitivity): The capacity of the model to identify all real anomalies, indicating how well the model can identify true positives.
- F1-Score: The symmetric means of precision and recall, ensuring that false positives and false negatives are balanced.
- AUC-ROC Curve: The trade-off between true positive and false positive rates at various thresholds is displayed by the area under the receiver operating characteristic curve.

4.7. Comparative Analysis

In this module, we compare the effectiveness of the standard Transformer model and the Transformer model augmented with clustering techniques. This comparison highlights the impact of incorporating clustering into the transformer-based anomaly detection approach.

- Transformer Model: The standard Transformer model utilizes only the attention mechanism to grasp the representations of the data without any explicit clustering of input features or latent space. It relies solely on its capacity to record long-term dependencies and contingent connections through multi-head attention and feed-forward layers.
- Transformer Model with Clustering Technique: In contrast, the combined model incorporates a clustering technique that groups the data into clusters before passing it through the Transformer’s architecture. This pre-processing step helps the model by enhancing the feature representations for different clusters of data, allowing the Transformer to focus more effectively on the underlying patterns of both normal and anomalous data inside each cluster.

4.8. Discussion

The results demonstrate that the Transformer-based neural network consistently outperforms the baseline models across all key metrics. Specifically:

- The precision and recall of the proposed model were significantly higher compared to the standard transformer model, indicating better detection of true anomalies while holding false positives to a minimum.
- The F1-score of the transformer model was also superior, demonstrating improved recall and accurate balancing, especially in datasets with imbalanced class distributions.

- The AUC-ROC result of the hybrid model consistently achieved higher values compared to both the standard transformer and traditional models, reflecting improved sensitivity in differentiating between normal and anomalous cases.

Overall, the integration of clustering techniques enhanced the ability of the transformer model to recognize underlying patterns in the data, which led to better anomaly detection performance. This hybrid approach also resulted in more interpretable results, as the clustering step provided additional insight into the nature of the anomalies detected.

5. Conclusion

This study highlights the effectiveness of combining clustering techniques with a transformer-based model for anomaly detection, significantly improving efficiency with respect to accuracy, precision, and recall. By grouping similar data points before applying the transformer, the hybrid model demonstrates a superior ability to detect anomalies in complex and imbalanced datasets. The results suggest that this approach enhances the robustness and sensitivity of anomaly detection systems, making it particularly valuable for critical applications such as cybersecurity and fraud detection.

Keeping an organizational network safe from financial, reputational, and legal risks has made defensive security a primary priority. Once compromised, networks and systems can be exploited to turn risk into an attack and exploit weaknesses. This study explores the effectiveness of our proposed Network Anomaly detection system using the Transformer Neural Networks and Clustering techniques. According to the results, the suggested ensemble classifier outperformed the other deep learning methods with an accuracy of over 99.5%, 98.5%, and 99.9% on the UNSW-NB15 dataset. Furthermore, the benchmark datasets showed good performance from the particular algorithms that were chosen.

Future research can explore more efficient clustering algorithms and techniques to improve real-time detection, such as using lighter transformer models. Automated hyperparameter tuning and transfer learning could enhance model optimization and generalization. Further work should also focus on improving the handling of imbalanced data and integrating explainability techniques to make the model’s decisions more transparent. Expanding the model’s capacity to process multimodal data could make it more adaptable to various real-world applications.

References

[1] N. Saran and N. Kesswani, “A comparative study of supervised Machine Learning classifiers for Intrusion Detection in Internet of Things”, *Procedia Computer Science,* vol. 218, no. 8, pp. 2049-2057, 2023.

[2] H. Lv and Y. Ding, “A hybrid intrusion detection system with K-Means and CNN+LSTM”, *EAI Endorsed Transactions on Scalable Information Systems,* vol. 11, no. 6, pp. 1-9, 2024.

[3] Y. Zhu, Y. Wang, L. Zhou, and Y. Xia, “FC-Trans: Deep Learning Methods for Network Intrusion Detection in Big Data Environments”, *Computers & Security,* vol. 154, 104392, 2025.

[4] X. Zhang, F. Yang, Y. Hu, Z. Tian, W. Liu, Y. Li, and W. She, “RANet: Network intrusion detection with group-gating convolutional neural network”, *Journal of Network and Computer Applications,* vol. 198c, 103266, 2022.

[5] T. Acharya, A. Annamalai, and M. Chouikha, “Enhancing the Network Anomaly Detection using CNN-Bidirectional LSTM Hybrid Model and Sampling Strategies for Imbalanced Network Traffic Data”, *Advances in Science Technology and Engineering Systems Journal,* vol. 9, no. 1, pp. 67-78, 2024.

[6] Y. Zhou, H. Zeng, Z. Zheng, and W. Zhang, “Network traffic anomaly detection model based on feature grouping and multi-autoencoders integration”, *Electronics Letters,* vol. 60, no. 23, 2024.

[7] Z. Long, H. Yan, G. Shen, X. Zhang, H. He, and L. Cheng, “A Transformer-based Network Intrusion Detection Approach for Cloud Security” *Journal of Cloud Computing,* vol. 13, no. 1, pp. 1-11, 2024.

[8] S. Tuli, G. Casale, and N.R. Jennings, “TranAD: deep transformer networks for anomaly detection in multivariate time series data”, *Proceedings of the VLDB Endowment,* vol. 15, no.

6, pp. 1201-1214, 2022.

[9] N. Al-Dahoul, H. Abdul Karim, and A.S.B. Wazir, “Model fusion of deep neural networks for anomaly detection”, *Journal of Big Data,* vol. 8, no. 106, 2021.

[10] H. Benaddi, K. Ibrahimi, A. Benslimane, and J. Qadir, “A Deep Reinforcement Learning Based Intrusion Detection System (DRL-IDS) for Securing Wireless Sensor Networks and Internet of Things”, *Lecture Notes of the Institute for Computer Sciences,* pp. 73-87, 2020.

[11] N. Kunhare, R. Tiwari, and J. Dhar, “Intrusion detection system using hybrid classifiers with meta-heuristic algorithms for the optimization and feature selection by genetic algorithm”, *Computers and Electrical Engineering,* vol. 103, no. 8, 2022.

[12] M. Maazalahi and S. Hosseini, “A Novel Hybrid Method Using Grey Wolf Algorithm and Genetic Algorithm for IoT Botnet DDoS Attacks Detection”, *Int J Comput Intell Syst,* vol. 18, no. 61, 2025.

[13] A. Davahli, M. Shamsi, and G. Abaei, “Hybridizing genetic algorithm and grey wolf optimizer to advance an intelligent and lightweight intrusion detection system for IoT wireless networks”, *J Ambient Intell Human Comput,* vol. 11, pp. 5581-5609, 2020.

[14] A.A. Diro, S. Kaisar, A.V. Vasilakos, A. Anwar, A. Nasirian, and G. Olani, “Anomaly Detection for Space Information Networks: A Survey of Challenges, Techniques, and Future Directions”, *Computers and Security,* vol. 139, 103705, 2024.

[15] S. Imran Ali, M.F. Zuhairi, S.M. Ali, Z. Shahid, M.M. Alam, and M.M. Su’ud, “Improving Reliability for Detecting Anomalies in the MQTT Network by Applying Correlation Analysis for Feature Selection Using Machine Learning Techniques”, *Applied Sciences,* vol. 13, no. 11, 2023.