5G-NIDD: A Comprehensive Analysis of Network Intrusion Detection Algorithms on 5G Wireless Network Dataset

Adit Sandeep Virkar*, Arnav Raviraj*, Vinay Kantilal Chavhan*, Shivanjay Vilas Wagh*
*Masters in Software Engineering
Arizona State University, Tempe, Arizona
Emails: avirkar@asu.edu, araviraj@asu.edu, vchavhan@asu.edu, swagh5@asu.edu

***Abstract*—This study investigates the escalating security challenges and vulnerabilities presented by the advancement of fifth-generation (5G) wireless networks. As these networks enhance connectivity, data rates, and device integration, they also introduce sophisticated threats due to their complex infrastructure and the integration of artificial intelligence (AI) and machine learning (ML). This research emphasizes the necessity of robust, comprehensive datasets like 5G-NIDD (5G Network Intrusion Detection Dataset) to effectively deploy AI and ML solutions for proactive threat detection and mitigation. By examining eight significant papers, this paper explores various facets of network security, from developing intrusion detection systems to applying AI for enhanced security protocols and exploring novel security challenges within 5G frameworks. The outcomes underscore the critical role of AI and ML in addressing modern network security challenges and set a foundation for future research and security measures in the rapidly evolving landscape of wireless technologies.**

***Index Terms*—Network Security, Cyber Threats, Intrusion Detection, Machine Learning, Artificial Intelligence, 5G Network.**

I. Introduction

With the advent of fifth-generation (5G) wireless networks and the nascent development of sixth-generation (6G) technologies, the complexity and sophistication of network infrastructures have significantly increased. This evolution brings unparalleled opportunities for enhanced connectivity, higher data rates, and seamless integration of various devices and applications. However, it also introduces new vulnerabilities and security challenges, primarily due to the expanded attack surface and the incorporation of artificial intelligence (AI) and machine learning (ML) technologies. These advancements necessitate reevaluating security measures and developing innovative solutions to safeguard future networks against cyber threats[2].

Integrating AI and ML in network security offers promising proactive threat detection and mitigation avenues. However, the effectiveness of these AI/ML-based security solutions is contingent upon the availability of comprehensive and relevant datasets that accurately reflect the complexity and dynamism of modern networks. Additionally, the transition to 6G networks, characterized by their intelligent and highly interconnected nature, further amplifies these security concerns, necessitating advanced security mechanisms capable of addressing the sophisticated threats in these environments.

In response to this pressing need, this paper presents the culmination of extensive research efforts to address the dearth of authentic datasets in AI-based 5G security. Leveraging a fully functional 5G network testbed, we use the 5G-NIDD—a pioneering network intrusion detection dataset meticulously crafted from real-world 5G network flows. Distinguished by its authenticity and fidelity to actual network dynamics, 5G-NIDD represents a significant advancement over existing datasets, offering researchers a rich resource for testing and validating novel intrusion detection algorithms[1].

In the subsequent sections, we elucidate the genesis, characteristics, and utility of 5G-NIDD and comprehensively evaluate its efficacy using diverse ML techniques. By bridging the gap between theoretical research and practical application, this endeavor aims to galvanize advancements in AI-based security within the ambit of B5G and 6G networks, thereby fortifying the foundations of future wireless communication infrastructures.

II. Literature survey

This literature survey delves into the current state of research on network security in the context of 5G and 6G technologies. It encompasses a review of eight pivotal papers exploring various aspects of network security, including developing comprehensive intrusion detection datasets, integrating AI for enhanced security measures, and exploring novel security challenges and solutions in the era of 5G and 6G networks.

The 5D-NIDD dataset underscores the complexity of securing 5G networks against sophisticated attacks. The dataset, developed in Oulu, Finland, aims to address the shortcomings of existing datasets by providing a rich source of real-world network scenarios, including various attack and benign traffic scenarios. This dataset is instrumental in applying ML techniques to develop proactive security measures against threats like Denial of Service (DoS) attacks and Port Scans, which are increasingly prevalent in 5G networks [1].

The transition towards 6G networks brings about a paradigm shift in network architecture, emphasizing the need for intelligent security mechanisms. The integration of AI into 6G security, as discussed in the second paper, offers a proactive

approach to threat detection and mitigation. This paper highlights the importance of AI in achieving connected intelligence and outlines the security challenges inherent in 6G's denser deployments and reliance on AI. The proposed hierarchical security mechanisms and blockchain integration offer a blueprint for securing future networks [2]. Moreover, it provides a forward-looking perspective on 6G wireless systems and emphasizes the need for innovation in network design, application focus, and performance measurement. The recommendations for utilizing high frequencies, reimagining network architecture, and leveraging intelligent surfaces provide a comprehensive framework for developing 6G technologies [3]. The potential of Federated Learning (FL) as a defense mechanism against adversarial attacks in 5G networks. By proposing novel defense strategies and emphasizing the importance of collaborative research, this paper contributes valuable insights into securing federated learning environments against sophisticated threats [4]. Also, focusing on the SliceSecure model addresses the unique challenges posed by network slicing in 5G architectures. The study uses deep learning techniques to offer a nuanced understanding of DDoS attacks within network slices. It proposes effective detection mechanisms, underscoring the importance of advanced intrusion detection systems in softwarized networks [5]. A comprehensive review of intrusion detection systems (IDS) highlights the evolution of detection methodologies in the face of sophisticated cyber-attacks. The discussion on the need for robust datasets and the exploration of evasion techniques offers a critical examination of the current landscape of IDS [6]. The empirical analysis compares the effectiveness of XGBoost against other machine learning classifiers. The findings underscore the efficiency and scalability of XGBoost in machine learning tasks, contributing to the broader discourse on applying ML techniques in cybersecurity [7]. Suppose we envision a 6G network architecture empowered by AI, detailing the strategic plans and technological innovations necessary for its realization. The emphasis on AI's role in enhancing situational awareness and enabling new service types highlights the transformative potential of integrating AI into wireless networks [8].

Furthermore, the survey explores recent advancements in AI-based intrusion detection and prevention techniques tailored to the unique characteristics of 5G and 6G networks. By leveraging AI and ML algorithms, researchers aim to develop proactive security measures capable of discerning subtle anomalies amidst the deluge of network traffic. These innovative approaches promise to fortify network defenses against emerging threats, including sophisticated attacks targeting critical infrastructure and sensitive data. Moreover, the survey examines the evolving landscape of threat vectors and attack methodologies, shedding light on the need for adaptive security mechanisms to thwart evolving cyber threats. Through a comprehensive analysis of recent research endeavors, this survey provides a holistic view of the evolving security challenges and mitigation strategies pertinent to the next-generation wireless ecosystem[9].

Conclusively, the surveyed literature underscores the critical role of AI and ML in addressing the security challenges of 5G and 6G networks. From developing comprehensive intrusion detection datasets to exploring AI-driven security mechanisms, the research efforts highlighted in this survey offer valuable insights and directions for future work in securing next-generation wireless networks. As we advance toward implementing 6G technologies, integrating intelligent security measures will be paramount in protecting against the evolving landscape of cyber threats. This literature survey, encompassing various research perspectives and methodologies, sets the stage for continued innovation and collaboration in network security.

III. Attack Types

This section describes the attack types considered in the generation of the 5G-NIDD. We have considered these types of attacks:

A. DOS attacks

A Denial of Service (DoS) attack is a malicious attempt to disrupt the normal functioning of a targeted server, service, or network by overwhelming it with a flood of illegitimate traffic or resource requests. The objective of a DoS attack is to render the target inaccessible to legitimate users, causing service degradation or complete outage.

1) ICMP Flood: An ICMP flood is a Denial of Service (DoS) attack in which an attacker sends a large volume of ICMP (Internet Control Message Protocol) packets to a target, overwhelming its network capacity and making it unreachable to legitimate users.

2) UDP Flood: A UDP flood is a DoS attack in which the attacker floods a target server with User Datagram Protocol (UDP) packets, consuming its network bandwidth and resources and leading to a denial of service for legitimate users.

3) SYN Flood: A SYN flood is a DoS attack where the attacker floods a target server with many TCP SYN (synchronization) packets, exhausting the server's resources and preventing it from servicing legitimate connection requests.

4) HTTP Flood: An HTTP flood is a DoS attack where the attacker floods a web server with HTTP requests, overwhelming its capacity to respond to legitimate users and causing the website to become unreachable or slow to respond.

5) Slowrate DOS: A slow-rate denial of service (DoS) attack aims to exhaust server resources by establishing connections with the target server and sending legitimate-looking requests at an extremely slow rate. This can tie up server resources, preventing it from responding to legitimate requests.

B. Port Scans

Port scanning probes a computer system or network to discover which ports are open and what services are running on those ports. Attackers commonly use port scan attacks to gather information about potential vulnerabilities or services running on a target system.

1) SYN Scan: A SYN scan is a port scanning technique where the attacker sends a SYN packet to the target server's ports to determine which ports are open, closed, or filtered. This can be used to identify potential vulnerabilities or services running on the target system.

2) TCP Connect Scan: A TCP connect scan is a port scanning technique in which the attacker attempts to establish a full TCP connection with the target server's ports to determine their status (open, closed, or filtered). This method is less stealthy than SYN scanning but provides more reliable results.

3) UDP Scan: A UDP scan is a port scanning technique where the attacker sends UDP packets to the target server's ports to determine their status. Since UDP is connectionless, this method is often used to scan for open UDP ports, which various network services and protocols may use.

In summary, the attack types considered in generating the 5G-NIDD dataset encompass a range of malicious activities to disrupt the normal operation of targeted servers, services, or networks. Denial of Service (DoS) attacks, including ICMP floods, UDP floods, SYN floods, HTTP floods, and Slowrate DOS attacks, aim to overwhelm network resources and render the target inaccessible to legitimate users. These attacks exploit vulnerabilities in network protocols and server capacity, causing service degradation or complete outage. Additionally, port scan attacks, such as SYN scans, TCP connect scans, and UDP scans, are employed by attackers to probe target systems for open ports and running services, facilitating reconnaissance and potential exploitation of vulnerabilities. The diversity of attack types underscores the multifaceted nature of network security challenges in the context of 5G networks, highlighting the importance of robust intrusion detection and prevention mechanisms to safeguard against evolving threats.

IV. Data processing

This section provides a detailed description of the various steps involved in data processing, including removing null values, feature selection, data normalization, and hyperparameter tuning.

A. Data Preprocessing

The 5G Network Intrusion Detection Dataset (5G-NIDD), derived from a 5G wireless network in Oulu, Finland, is valuable for identifying patterns and potential security vulnerabilities within 5G networks. In this study, we utilized two primary files from the dataset: combined.csv and encoded.csv, which respectively contained concatenated data for all attack scenarios for both base stations as network flows. During the data cleaning process, we focused on removing all null values, a critical step in preparing the data for in-depth correlation analysis. This step was paramount to ensuring the integrity and accuracy of the analysis, as null values can significantly skew results and lead to misleading conclusions. By enhancing the dataset's quality through rigorous data cleaning, we established a solid foundation for generating insightful and accurate findings for detecting network intrusions in 5G networks. For the data labeling process, we labeled the data into two types, namely, Benign and Malicious, for binary classification and into nine types, namely, Benign, ICMP Flood, UDP flood, SYN flood, HTTP flood, Slow rate DoS, SYN Scan, TCP Connect Scan, and UDP Scan for multiclass classification. Since there is no use of categorical data in Machine Learning, we converted them into numerical characters using one hot encoding technique. One-hot encoding is a technique used to convert categorical variables into binary vectors, where each category is represented by a binary value, typically 1 or 0, indicating its presence or absence [1].

B. Feature Selection and Extraction

Feature Selection is a fundamental step in Machine Learning as it reduces the dimensionality of the data, improves the model's performance, enhances the interpretability as the model's complexity decreases, and reduces the training time. To achieve this, it is essential to identify the relevant features of the model and select the highly correlated features. We have mainly implemented two types of correlation coefficients: Pearson and Spearman. We can comprehensively analyze the relationships between various features within the dataset based on the Pearson and Spearman correlation coefficients obtained [8].

1) Pearson Correlation: The Pearson correlation coefficient measures the linear correlation between two variables, giving a value between -1 and 1. A value closer to 1 implies a strong positive correlation, while a closer to -1 indicates a strong negative correlation. A value around 0 suggests no linear correlation. We analyzed the correlation between Seq (sequence number) and most other features, which is close to 0, indicating that sequential data points do not linearly predict the behavior of different features in the dataset [8]. Moreover, a perfect correlation (1.0) among Dur, Runtime, Mean, and Sum indicates these features move together linearly. This might suggest redundancy among these features, which could be condensed or simplified in predictive modeling to improve efficiency without losing critical information [8].

2) Spearman Correlation: The Spearman correlation assesses the monotonic relationship between two variables, which is more flexible than the Pearson correlation as it can capture relationships that are not strictly linear. The Spearman correlation coefficients highlight similar trends to the Pearson correlation, with most features exhibiting low to moderate correlation. However, the Spearman correlation identifies stronger monotonic relationships, which could be either linear or nonlinear but consistently increasing or decreasing, whereas the Pearson correlation did not. This discrepancy can indicate the presence of nonlinear relationships among certain features [8]. For example, Rate, SrcRate, and DstRate have significantly different Spearman coefficients than their Pearson counterparts, suggesting that their relationships might be more complex and nonlinear [8].

After analyzing the correlation among the columns, we have aimed to identify the columns in the 5G-NIDD dataset that are highly correlated with each other based on a specific correlation threshold. We have set the threshold to 0.9 for

TABLE I
HIGHLY CORRELATED COLUMNS IDENTIFIED BY PEARSON AND SPEARMAN CORRELATION

Spearman	Pearson	Uncommon Columns
Runtime	Mean	dMeanPktSz
DstBytes	TotPkts	Loss
Rate	Load	TcpRtt
dMeanPktSz	RunTime	AckDat
Loss	SrcRate	pLoss
TotBytes	Sum	SynAck
TotPkts	DstPlts	
TcpRtt	Dur	
DstPkts	Max	
Dur	SrcPkts	
Max	DstBytes	
SrcPkts	DstLoad	
SynAck	SrcLoad	
SrcBytes	Rate	
AckDat	SrcBytes	
Min	Min	
pLoss		
Mean		
Load		
SrcRate		
Sum		
DstRate		
SrcLoad		

both Pearson and Spearman. As a result, Pearson identified 18 different columns, and Spearman recognized 24 different columns that are highly correlated (Table 1). Spearman's method identified more correlated pairs, suggesting that some relationships between variables in the provided dataset are monotonic but not linear. Spearman correlation can capture these relationships because it is based on rank values rather than the raw data, making it sensitive to monotonic trends, whether linear or not. The columns identified exclusively through Spearman but not Pearson ("Uncommon") exhibit strongly monotonic relationships but are not strictly linear. This indicates variability in how different features relate, suggesting that some features might have a nonlinear impact on others, which Pearson's correlation fails to capture due to its focus on linear relationships [8]. The "Uncommon" columns (e.g., dMeanPktSz, Loss, TcpRtt, etc.) are fascinating because they were only highlighted by Spearman correlation. This suggests that these features have a robust and consistent relationship with at least one other feature in a nonlinear way. These features could be crucial in specific analysis scenarios, especially in understanding complex, nonlinear dynamics within 5G-NIDD [8].

After conducting correlation analysis with Pearson and Spearman coefficients, we further evaluated the importance of features using the Random Forest algorithm. This approach provides additional insights into the predictive power of features by measuring their impact on the model's accuracy. Random Forest, an ensemble learning method, offers a straightforward metric to evaluate feature importance. It does so by observing how random permutations of each feature affect the model's performance, thereby indicating the significance of each feature in predicting the target variable. We applied Random Forest to extract the top 10, 15, and 25 features based on their importance scores. This process corroborated some of our findings from the correlation analysis and revealed additional features that significantly influence model prediction yet might not have shown strong linear or monotonic relationships. Selecting features based on the importance of Random Forest allows for a more nuanced approach to reducing dimensionality. It ensures that the features included in the model encapsulate the most relevant information for predicting network intrusions in 5G networks, thus enhancing model performance and interpretability while reducing training time. By leveraging correlation analysis and the importance of the Random Forest feature, we have developed a robust feature selection process that combines the strengths of statistical analysis and machine learning algorithms. This comprehensive approach ensures that our model is built on a foundation of the most relevant and impactful features, optimizing its ability to detect network intrusions within the 5G-NIDD dataset.

A Pearson Correlation Heatmap visually represents the linear relationships between pairs of variables, quantified by the Pearson correlation coefficient Fig. 1. This coefficient ranges from -1 to +1, where values close to +1 or -1 indicate strong positive or negative linear relationships, respectively, and values around 0 suggest no linear relationship. Analyzing a dense heatmap with numerous features can be challenging. Still, patterns are discernible, such as blocks of dark red squares representing variables with a strong positive correlation, moving together linearly. This type of analysis is crucial in fields like machine learning, where understanding correlations helps in feature selection and reducing redundancy, potentially averting issues like overfitting.

On the other hand, a Spearman Correlation Heatmap measures the monotonic relationships between variables, which do not necessarily have to be linear, using ranked values instead of raw data Fig. 2. While similar in appearance to the Pearson heatmap, the Spearman heatmap might highlight different relationships, reflecting the non-linear or ordinal nature of the data. This can be particularly useful when the data does not meet the assumptions required for Pearson correlation, allowing for a broader exploration of how variables relate. Both heat maps typically show perfect correlation along the diagonal (since each variable is correlated with itself) and are symmetric, indicating that correlation is mutual regardless of the variable order.

Navigating these complex heatmaps might require interactive visualization tools or simplifying the data by focusing on specific variables or clustering similar ones. The insights gained from these correlations are instrumental in various analytical practices, from reducing dimensionality with techniques like PCA to tailoring feature selection for predictive models. Understanding and addressing potential issues such as multicollinearity, evident from highly correlated variables, can enhance the robustness and interpretability of model outcomes.

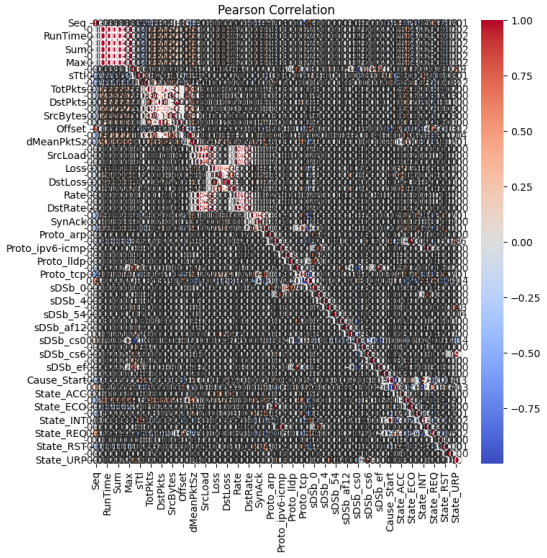

Fig. 1. Pearson Heatmap

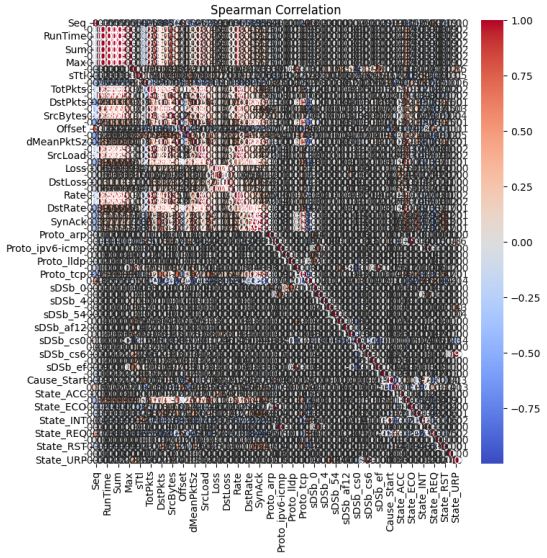

Fig. 2. Spearman Heatmap

C. Data Normalization

Data Normalization is essential as the input variables need to be scaled to a standard range without distorting differences between the ranges of values. This is particularly important because machine learning algorithms that use distance metrics (such as k-nearest neighbors (KNN), k-means clustering, or gradient descent algorithms like linear regression and neural networks) are sensitive to the magnitude of the data. Without normalization, a feature with a broader range could disproportionately influence the model, leading to biased results. We have incorporated Min-Max scaling, which transforms features to have a specific minimum and maximum, which can be crucial for models sensitive to data variance. We used (1) to calculate the min-max data normalization.

TABLE II
RANKING OF TOP 10, 15, AND 25 FEATURES BASED ON RANDOM FOREST

Top 10 Features	Top 15 Features	Top 25 Features
Seq	Seq	Seq
Offset	Offset	Offset
sTtl	sTtl	sTtl
sHops	sHops	sHops
sMeanPktSz	sMeanPktSz	sMeanPktSz
SrcBytes	SrcBytes	SrcBytes
Proto_tcp	Proto_tcp	Proto_tcp
TotBytes	TotBytes	TotBytes
TcpRtt	TcpRtt	TcpRtt
State_REQ	State_REQ	State_REQ
	SyncAck	SyncAck
	AckDat	AckDat
	SrcRate	SrcRate
		SrcLoad
		Cause_Start
		Cause_Status
		Proto_udp
		State_ECO
		State_INT
		Load
		Rate
		Mean
		Min
		Dur
		DstPkts

$$X_{\text{scaled}} = \frac{X - \min(X)}{\max(X) - \min(X)} \tag{1}$$

X is the original value, and Xscaled is the scaled value. Also, Xmin and Xmax are the maximum and maximum of the feature across all the data points. After applying the formula for each value of each feature to the Xscaled formula, we transform the original feature values with their scaled versions. This helps optimize the performance and enhances the effectiveness of the machine learning algorithms.

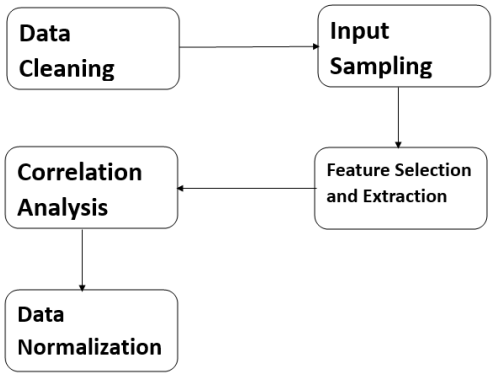

Fig. 3. Data Processing

D. File Description

The 5G-NIDD datasets in the SER_517_F_33 GitHub repository are essential for analyzing Network-Initiated Dedicated Device performance within 5G networks. The repository

includes a variety of crucial files such as raw data, processed datasets, summary statistics, and model inputs. These files cover everything from the initial raw data, which captures metrics such as signal strengths and device identifiers, to fully processed datasets that are prepared for modeling and include engineered features and cleaned data. Each file facilitates different data analysis and modeling stages to improve network design and enhance user experience in 5G settings.

Researchers accessing these datasets will find detailed summary statistics and specifically tailored features conducive to advanced analysis and predictive modeling. Hosting these datasets on GitHub enables collaborative enhancements and broadens accessibility, leading to deeper insights into 5G technology.

V. Analysis and Discussion

This section provides a detailed analysis of the dataset using various machine learning (ML) models, including deep neural networks. It describes the performance metrics used to evaluate the models individually and specifies the ML models employed. The section concludes with a comparative analysis of the performance of each ML model for both binary classification tasks.

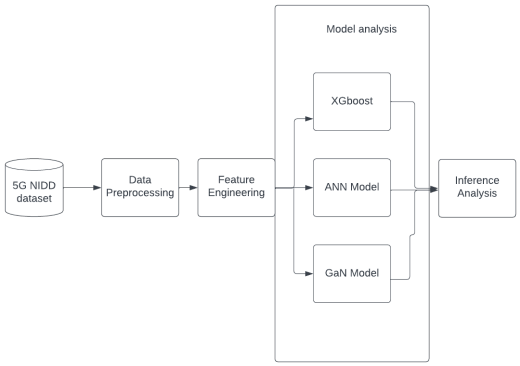

Fig. 4. Workflow of our ML models

A. Performance Measuring Metrics

We have used the most common performance measuring metrics, including the confusion matrix, precision, recall, F-1 score, and accuracy.

1) Confusion Matrix: A confusion matrix is a table that visualizes the performance of a classification model by comparing actual and predicted values. It consists of four components: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

TABLE III
Confusion matrix for Binary Classification

	Actual Positive Class	**Actual Negative Class**
Predicted Positive Class	True Positive (TP)	False Negative (FN)
Predicted Negative Class	False Positive (FP)	True Negative (TN)

2) Precision: Precision measures the accuracy of positive predictions made by the model. It is calculated as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Where TP is the number of true positive predictions, and FP is the number of false positive predictions.

3) Recall: Recall measures the ability of the model to capture all positive instances. It is calculated as:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Where TP is the number of true positive predictions, and FN is the number of false negative predictions.

4) F-1 score: The f-1 score is the harmonic mean of precision and recall, balancing the two metrics. It is calculated as:

$$\text{F-1 Score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}$$

5) Accuracy: Accuracy measures the overall correctness of the model's predictions across all classes. It is calculated as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Where TP is the number of true positive predictions, TN is the number of true negative predictions, FP is the number of false positive predictions, and FN is the number of false negative predictions.

6) Accuracy: Accuracy measures the overall correctness of the model's predictions across all classes. It is calculated as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Where TP is the number of true positive predictions, TN is the number of true negative predictions, FP is the number of false positive predictions, and FN is the number of false negative predictions.

B. Boosting models

1) XGBoost: The provided Python code leverages the XGBoost (Extreme Gradient Boosting) library for a classification task. Initially, it prepares the dataset by removing irrelevant columns, handling missing values, and encoding categorical variables. Subsequently, the code splits the data into training and testing sets, where an XGBoost classifier model is initialized and trained on the training data. Following training, the model predicts labels for the test data and calculates the accuracy score to assess its performance. Additionally, the code generates a confusion matrix to provide a detailed evaluation of the model's ability to classify instances correctly. Optional sections for parameter tuning using grid search are included to address overfitting by exploring different combinations of hyperparameters. Overall, this code exemplifies the practical implementation of XGBoost for classification tasks, encompassing data preprocessing, model training, evaluation, and potential optimization through parameter tuning.

2) AdaBoost: We implemented the AdaBoostClassifier algorithm from the sci-kit-learn library. We started by preprocessing the data, including handling categorical variables using one-hot encoding and filling null values with zeros. The dataset is then split into training and testing sets. The AdaBoostClassifier is instantiated with 1000 estimators and trained on the training data. After training, predictions are made on the test set, and a classification report is generated to evaluate the model's performance, including precision, recall, and F1-score for each class. Additionally, the accuracy and training time is recorded for performance analysis.

3) CatBoost: The provided Python code showcases the utilization of CatBoost, a powerful machine-learning library tailored for handling categorical data efficiently. The code begins by preparing the dataset, including dropping irrelevant columns, imputing missing values, and encoding the target variable using LabelEncoder. It then splits the data into training and testing sets and initializes a CatBoostClassifier with specified parameters for model training. The CatBoost classifier is trained on the training data, with training progress printed every 100 iterations. Subsequently, the trained model is used to predict labels for the test data, and the model's accuracy is computed. A confusion matrix is also generated to evaluate the model's performance, providing insights into its ability to classify instances correctly. This code exemplifies a comprehensive workflow for utilizing CatBoost in a classification task, encompassing data preprocessing, model training, prediction, and performance evaluation.

C. Artificial Neural Network

A fully connected artificial neural network has been implemented to model the data. The network architecture consists of an input layer, three hidden layers, and an output layer. The hidden layers are composed of densely connected neurons, allowing for complex nonlinear transformations and feature extraction. The rectified linear unit (ReLU) activation function is employed in the hidden layers to introduce non-linearity and enable the network to learn intricate patterns. The Adam optimizer, a state-of-the-art adaptive learning rate optimization algorithm, has been chosen to train the network efficiently. The Adam optimizer adaptively updates the learning rate for each weight based on the gradients, providing faster convergence and improved generalization performance compared to traditional stochastic gradient descent methods. Currently, the model is being trained using the specified architecture and optimizer to minimize the defined loss function and achieve accurate predictions on the target variable. More research is needed to find the correct number of hidden layers and to avoid overfitting. Another Artificial Neural Network (ANN) implementation uses TensorFlow and Keras for binary classification. It begins with data preprocessing steps, including handling missing values, normalization, and label encoding. The ANN architecture consists of four fully connected layers with ReLU activation in hidden layers and sigmoid activation in the output layer for binary classification. It employs a Stochastic Gradient Descent (SGD) optimizer with a learning rate 0.01 and a Sparse Categorical Cross-entropy loss function. The model is trained on the training data with a batch size of 32 for ten epochs, with a validation split of 10

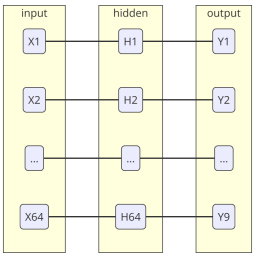

Fig. 5. Simple Ann Architecture

D. GANBased Network

The generator network creates new data samples, such as attack rows, by transforming random noise into realistic-looking outputs. On the other hand, the discriminator network tries to distinguish between actual data samples and those generated by the generator. Through this adversarial process, both networks improve over time, with the generator learning to produce more convincing samples and the discriminator becoming better at distinguishing real from fake. In this research, GANs have been used to create synthetic 5G network data, including attacks. It has shown remarkable capabilities in producing high-quality, realistic outputs, sometimes indistinguishable from accurate data. [10]

1) GAN Architecture: The GAN consists of a generator and a discriminator. The discriminator takes the real and generated data samples as input and tries to distinguish between them. The generator takes random noise as input and tries to generate synthetic data resembling real data. The GAN combines the generator and discriminator to train them simultaneously in an adversarial manner.

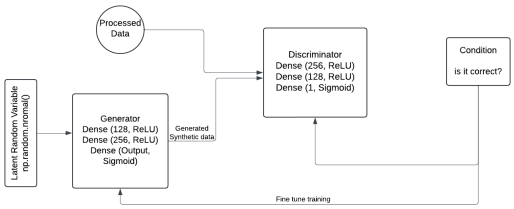

Fig. 6. Architecture of GAN-based model

2) Training the GAN: The GAN is trained in a loop for 1000 epochs with 64 batch sizes. In each epoch, synthetic data is generated by the generator using random noise. Real data samples are randomly selected from the training processed dataset. The discriminator is trained using real and synthetic data, with labels indicating whether the data is real or fake by determining whether it's attack or being transmission. The generator is trained through the GAN model to generate data that can fool the discriminator into predicting it as real.

3) Generating Synthetic Data: Synthetic data is generated using the trained generator having three dense layers after training the GAN.

4) Downstream Classification Task: The generated synthetic data is combined with the original training data.

A Classifier such as CatBoost is trained on the combined processed dataset, including real and synthetic data. The classifier is trained to predict the binary labels (benign or malicious) based on the input features. The classifier's accuracy is evaluated on the test data after training.

5) Accuracy Evaluation: The accuracy of the classifier on the synthetic data is computed using the test dataset. A confusion matrix is also generated to assess the classifier's performance regarding true positives, false positives, and false negatives. We used GAN to generate synthetic data to augment the training dataset for a downstream classification task. The classifier's accuracy on the synthetic data provides insights into the effectiveness of GAN-generated data for improving the performance of machine learning models.

E. Binary Classification

We performed binary classification on the dataset to distinguish between malicious and benign flows. Using an 80-20% train-test split of the total flows, we experimented with various features selected by Pearson and Spearman correlation. Our experiments also used the top 10, 15, and 25 features. Optimal performance in terms of accuracy and computational time was achieved with the top 10 features. Each experiment was repeated 5 times, and the average result was used as the final value presented in this paper.

TABLE IV
COMPARISON OF MACHINE LEARNING MODELS FOR ENCODED DATA

Model	Type	Precision	Recall	F-1 score	Accuracy	Training Time(s)	Prediction Time(s)
XGBoost	Benign	0.99	0.99	1.0	100%	50.26	1.24
	Malicious	0.99	1.00	0.99			
AdaBoost	Benign	0.99	0.98	0.99	99.56%	212.34	1.7
	Malicious	0.98	0.97	0.98			
CatBoost	Benign	0.99	0.99	0.99	99.98%	256.52	0.39
	Malicious	0.99	0.98	0.98			
ANN (PyTorch)	Benign	0.62	0.65	0.59	61%	5185.08	0.49
	Malicious	0.61	0.64	0.66			
ANN (TensorFlow)	Benign	0.65	0.66	0.61	63%	1853.67	0.54
	Malicious	0.61	0.64	0.57			
GAN	Benign	0.99	0.99	0.99	99.97%	75.77	0.16
	Malicious	0.99	0.98	0.98			

TABLE V
COMPARISON OF MACHINE LEARNING MODELS FOR PEARSON CORRELATION

Model	Type	Precision	Recall	F-1 score	Accuracy	Training Time(s)	Prediction Time(s)
XGBoost	Benign	0.98	0.97	0.98	98.5%	35.72	0.92
	Malicious	0.99	0.97	0.98			
AdaBoost	Benign	0.99	0.99	0.99	99.8%	170.56	1.4
	Malicious	0.99	0.99	0.99			
CatBoost	Benign	0.98	0.97	0.98	98.2%	210.23	0.32
	Malicious	0.99	0.97	0.98			
ANN (PyTorch)	Benign	0.61	0.64	0.60	60.5%	4000.89	0.42
	Malicious	0.60	0.60	0.57			
ANN (TensorFlow)	Benign	0.63	0.65	0.61	62%	1400.45	0.46
	Malicious	0.64	0.64	0.63			
GAN	Benign	0.98	0.97	0.98	98.4%	30.37	0.20
	Malicious	0.97	0.96	0.99			

F. Analysis of Results

The binary classification results were excellent for all machine learning models except for ANN. Metrics such as Accuracy, Precision, Recall, and F1-Score were almost identical for XGBoost, AdaBoost, CatBoost, and GAN, with slight variations. ANN consistently showed lower performance across all evaluation metrics. Among the models, XGBoost performed best in identifying malicious traffic with the fewest false negatives, while CatBoost had the lowest number of false positives.

TABLE VI
COMPARISON OF MACHINE LEARNING MODELS FOR SPEARMAN CORRELATION

Model	Type	Precision	Recall	F-1 score	Accuracy	Training Time(s)	Prediction Time(s)
XGBoost	Benign	0.99	0.99	1.0	100%	45.87	1.1
	Malicious	0.99	0.98	0.99			
AdaBoost	Benign	0.99	0.98	0.99	99.6%	200.78	1.6
	Malicious	0.98	0.96	0.98			
CatBoost	Benign	0.99	0.99	0.99	99.95%	235.45	0.36
	Malicious	0.97	0.99	0.98			
ANN (PyTorch)	Benign	0.62	0.65	0.59	61%	4900.67	0.45
	Malicious	0.63	0.66	0.54			
ANN (TensorFlow)	Benign	0.64	0.65	0.60	62.5%	1750.89	0.49
	Malicious	0.67	0.66	0.64			
GAN	Benign	0.99	0.99	0.99	99.96%	70.92	0.15
	Malicious	0.99	0.99	0.98			

Different models exhibited varying training and prediction times. XGBoost had the shortest training time, followed by GAN, CatBoost and AdaBoost. The Artificial neural network(ANN) required more time for training compared to other models. Prediction time was generally low for all models even for ANN, which had a considerably longer prediction time compared to its training time.

For Binary classification, accuracy levels were high for each attack type, except for ANN, which performed poorly. The other models detected attack types with similar accuracy levels. However, ANN is not suitable for these types of problems. A closer examination of the results revealed that different models excelled at detecting certain types of attacks. Overall, XGBoost had a slightly higher accuracy than the other models.

VI. CONCLUSION

The advancements in 5G and the nascent development of 6G technologies present exciting opportunities for enhanced connectivity, higher data rates, and seamless integration of various devices and applications. However, these advancements also pose significant challenges related to network security and the expanded attack surface, especially with the increased use of artificial intelligence (AI) and machine learning (ML) technologies.

In this paper, we conducted a comprehensive analysis and survey of the current state of network security research in the context of 5G technologies. The focus was on the importance of comprehensive datasets like 5G-NIDD for applying machine learning techniques to effectively detect and mitigate cyber threats. Through our evaluation, we identified key aspects such as the relevance of feature selection and extraction, data normalization, and the critical need for innovative security solutions.

Our exploration of various machine learning algorithms, including boosting models (XGBoost, AdaBoost, and CatBoost), artificial neural networks (ANN), and generative adversarial networks (GANs), provided valuable insights into their performance and applicability in network security tasks. Boosting models demonstrated superior accuracy and efficiency, with

higher accuracy and reduced training times compared to ANN and GANs. Moreover, boosting models exhibit greater resilience to adversarial attacks, which can critically impact the reliability and security of AI models in network environments.

Conversely, ANN and GANs were more susceptible to adversarial attacks, which can undermine their performance and security. Therefore, choosing the appropriate model involves weighing these trade-offs between efficiency, accuracy, and robustness.

In conclusion, the research presented in this paper highlights the need for ongoing innovation and collaboration in network security. By leveraging the potential of AI and ML and continuing to develop comprehensive intrusion detection datasets, we can create robust and proactive security solutions that safeguard future wireless networks against sophisticated and emerging cyber threats. The insights on boosting models as compared to ANN and GANs guide choosing the optimal trade-offs for securing next-generation wireless networks. This paper sets the stage for continued advancement in securing the wireless networks of tomorrow.

VII. Future Work

In light of the growing use of deep learning and neural network models across various domains, the threat of adversarial attacks poses significant challenges and risks to their reliability and safety. Adversarial attacks exploit the vulnerability of these models by subtly altering input data to deceive the network into making incorrect predictions or classifications. This can lead to undesirable consequences, especially in high-stakes applications such as autonomous vehicles, medical diagnostics, and financial systems. Although boosting models, which use ensemble methods like Gradient Boosting or AdaBoost, are less susceptible to adversarial attacks due to their layered approach and the distribution of decision-making across multiple models, it is crucial to address vulnerabilities in neural networks and deep learning to ensure their continued effectiveness and trustworthiness.

Future work in this area should focus on creating more resilient architectures and defense mechanisms against adversarial attacks. Researchers should explore advanced techniques for adversarial training, such as incorporating adversarial examples in the training process to improve model robustness. Additionally, it is essential to investigate the detection of adversarial inputs in real time, providing immediate alerts or automatic countermeasures when suspicious activity is detected. Collaboration across research communities and industries can foster the development of standardized evaluation frameworks to assess the resilience of neural network models and benchmark different defense strategies. Moreover, while boosting models have demonstrated more resistance to adversarial attacks, their robustness should continue to be explored and enhanced for broader applications. By prioritizing these research directions, we can strengthen the security and integrity of deep learning models and maintain public confidence in their widespread adoption.

References

[1] Sehan Samarakoon, Yushan Siriwardhana, Pawani Porambage, Madhusanka Liyanage, Sang-Yoon Chang, Jinoh Kim, Jonghyun Kim, Mika Ylianttila. (2022). 5G-NIDD: A Comprehensive Network Intrusion Detection Dataset Generated over 5G Wireless Network. arXiv preprint arXiv:2212.01298.

[2] Walid Saad, Mehdi Bennis, Mingzhe Chen. (2019). A Vision of 6G Wireless Systems: Applications, Trends, Technologies, and Open Research Problems. arXiv preprint arXiv:1902.10265.

[3] Khan, Md Sajid and Farzaneh, Behnam and Shahriar, Nashid and Saha, Niloy and Boutaba, Raouf. (2022). SliceSecure: Impact and Detection of DoS/DDoS Attacks on 5G Network Slices. 2022 IEEE Future Networks World Forum (FNWF).

[4] Khraisat, Ansam and Gondal, Iqbal and Vamplew, Peter and Kamruzzaman, Joarder. (2019). Survey of intrusion detection systems: techniques, datasets, and challenges. Cybersecurity.

[5] Letaief, Khaled B and Chen, Wei and Shi, Yuanming and Zhang, Jun and Zhang, Ying-Jun Angela. (2019). The Roadmap to 6G: AI Empowered Wireless Networks. IEEE Communications Magazine. August.

[6] Saeed, Mamoon M and Saeed, Rashid A and Abdelhaq, Maha and Alsaqour, Raed and Hasan, Mohammad Kamrul and Mokhtar, Rania A. (2023). Anomaly Detection in 6G Networks Using Machine Learning Methods. Electronics. 2023, 12, 3300.

[7] Bentéjac, Candice and Csörgő, Anna and Martínez-Muñoz, Gonzalo. A Comparative Analysis of XGBoost.

[8] Chen, Tianqi and Guestrin, Carlos. (2016). XGBoost: A Scalable Tree Boosting System.

[9] Alhajjar, Elie and Maxwell, Paul and Bastian, Nathaniel D. (2020). Adversarial Machine Learning in Network Intrusion Detection Systems. arXiv preprint arXiv:2004.11898.

[10] Xin Xu1, Hongbo Zhao1,2, Haoqiang Liu1, Hua Sun1. 2020. LSTM-GAN-XGBOOST Based Anomaly Detection Algorithm for Time Series Data.