Detecting Intrusion in Softwarized 5G Networks Using Machine Learning

Abdallah Moubayed
School of Computing and Augmented Intelligence
Arizona State University
Tempe, USA
abdallah.moubayed@asu.edu

Vignesh Ramesh
School of Computing and Augmented Intelligence
Arizona State University
Tempe, USA
vrames25@asu.edu

Poornima Sathya Keerthi
School of Computing and Augmented Intelligence
Arizona State University
Tempe, USA
pskeerth@asu.edu

Shanmugapriyan Ravichanthiran
School of Computing and Augmented Intelligence
Arizona State University
Tempe, USA
sravic23@asu.edu

Joseph Thomas
School of Computing and Augmented Intelligence
Arizona State University
Tempe, USA
jthom175@asu.edu

Prasant Ganesan
School of Computing and Augmented Intelligence
Arizona State University
Tempe, USA
pganesa2@asu.edu

***Abstract*—The emergence of 5G technology has transformed the wireless communication landscape, offering unprecedented data speeds and connectivity for numerous devices and applications. However, this technological advancement also introduces new cybersecurity vulnerabilities, making 5G networks an attractive target for sophisticated cyber-attacks. This paper addresses the critical need for robust intrusion detection systems (IDS) in softwarized 5G networks by leveraging advanced machine learning (ML) techniques. Utilizing the comprehensive Network Intrusion Detection Dataset (NIDD) generated over 5G wireless, available through IEEE Dataport, we propose and evaluate several complex ML models aimed at accurately classifying network activities as either malicious or benign. Our research contributes to the ongoing efforts to secure 5G networks by providing insights into the effectiveness of ML-based IDS in detecting a wide range of cyber threats, thereby ensuring the integrity and reliability of 5G services.**

***Index Terms*—5G Security, Intrusion Detection, Machine Learning**

I. Introduction

The introduction of 5G technology represents a significant advancement in wireless communication, offering substantial improvements in data speeds, latency reduction, and enhanced network capacity. This breakthrough technology enables innovative applications across various industries, such as autonomous vehicles, remote medical procedures, and the Internet of Things (IoT). However, the complexity and open nature of 5G networks also expose them to numerous security threats and vulnerabilities, making them attractive targets for cyber attackers. Traditional security measures are often insufficient to counter these evolving threats, necessitating the development of advanced intrusion detection systems (IDS).

In this context, machine learning (ML) techniques emerge as a powerful tool to enhance the detection capabilities of IDS in software-defined 5G networks. ML algorithms can learn and adapt to changing patterns of network traffic, identifying both known and unknown threats with high accuracy. This research focuses on applying ML techniques to the 5G Network Intrusion Detection Dataset (5G NIDD) [5], a comprehensive dataset specifically designed for network intrusion detection in 5G wireless environments. By developing and evaluating several complex ML models, we aim to demonstrate the potential of ML in effectively classifying network activities as malicious attacks or benign behavior.

The structure of our research is organized as follows: We begin by reviewing existing literature, focusing on the current security challenges faced by 5G networks and the role of machine learning (ML) techniques in addressing these challenges. Next, we provide a detailed description of the 5G Network Intrusion Detection Dataset (5G NIDD) and outline the methodology employed in developing our ML models. The subsequent sections present a comprehensive analysis of the experimental results, highlighting the performance of each ML model in accurately detecting various types of cyber threats. Finally, we discuss the implications of our findings for enhancing the security of 5G networks and propose potential directions for future research in this domain.

II. Related Works

A. 5G-NIDD: A Comprehensive Network Intrusion Detection Dataset Generated over 5G Wireless Network

In addressing the increasing complexity and sophistication of 5G networks, which inevitably introduces new vulnerabilities, the paper "5G-NIDD: A Comprehensive Network Intrusion Detection Dataset Generated over 5G Wireless Network" [6] underscores the necessity for a novel dataset specifically tailored for network intrusion detection within 5G networks. Highlighting the significant technological advancements and unique security challenges of 5G, the paper points out the inefficiency of existing threat detection mechanisms when applied to the 5G context. This shortfall establishes the groundwork for the creation of the 5G-NIDD dataset [5], which is meticulously designed to cater to the specific needs of 5G network security. Through a detailed methodology, the paper elaborates on the development of this dataset, presenting an in-depth analysis that includes statistical information and a diverse range of attack types. This comprehensive approach ensures the dataset's relevance for machine learning models, aiming to enhance the precision of intrusion detection in 5G networks. The effectiveness of the 5G-NIDD dataset is further demonstrated through the application of various machine learning algorithms, highlighting its potential to significantly improve detection accuracy. This exploration into the dataset's utility in identifying cyber threats within 5G networks not only validates its importance but also marks a critical step forward in the ongoing efforts to secure 5G infrastructures.

B. Distributed Learning-Based Intrusion Detection in 5G and Beyond Networks

As the complexity and sophistication of 5G-advanced and subsequent 6th generation mobile networks escalate, so too does the frequency and complexity of cyberattacks targeting these networks. In the realm of network security, there's a burgeoning emphasis on preemptive cyberattack detection through network intrusion detection systems (NIDS). Yet, a glaring limitation within the existing body of research is its predominant focus on centralized environments, which are increasingly incongruent with the distributed nature of contemporary network architectures. Addressing this critical gap, the paper titled "Distributed Learning-Based Intrusion Detection in 5G and Beyond Networks" [3] introduces an innovative distributed learning-based NIDS, adept at functioning within decentralized settings. Leveraging the principles of Split Learning, the proposed system enables distributed learning across a network's nodes, irrespective of their computational capacities, focusing specifically on the SplitNN model. This approach is meticulously validated through experiments conducted on the 5G-NIDD dataset [5], illustrating that Split Learning not only holds its ground in environments with uniform data distribution compared to centralized models but also maintains robust intrusion detection performance in scenarios marked by data imbalance. Such findings illuminate the efficiency and applicability of Split Learning in enhancing network intrusion detection within the distributed ecosystems of 5G and beyond, offering a promising avenue for safeguarding next-generation networks against the evolving landscape of cyber threats.

C. Critical Analysis of 5G Networks Traffic Intrusion using PCA, t- SNE and UMAP Visualization and Classifying Attacks

We have 62 features in our dataset, and we have to reduce the dimensions of the dataset to train and test the model effectively. One of the main challenges we face is selecting the essential features and how to prune the unimportant ones. This paper explains why dimensionality reduction will play a massive role in reducing and extracting features from the original dataset. Two essential steps for dimensionality reduction are feature selection and feature extraction. Mutual information plays a significant role in feature selection as it calculates information gain between the features and selects the top K features from the N features. After selecting K features, if there is still more than one feature, we have to perform feature extraction, which effectively utilizes all the available features and extracts significant data without any data leakage. PCA comes into play when we have to select X components. We chose X by using three different functions: scree plot, loadings, and calculating the total variance. Once we effectively reduce the number of features, we can train and test the X features compared to the initial N features.

D. Exploring Emerging Trends in 5G Malicious Traffic Analysis and Incremental Learning Intrusion Detection Strategies

This paper [7] effectively states various network attacks from 1G to 5G and explains how the attack rates have been increased, and intrusion detection systems are the need of the hour. 5G utilizes new technologies such as Software Defined Networking and Network Function Virtualization, creating a 5G network as an effective target for network attacks. This paper talks about the traditional methods of identifying an attack and how it is nearly impossible to detect it using traditional methods. Due to the advancement in technology, no two packets will be the same, because of this, machine learning methods need to be equipped for effective intrusion detection, and machine learning does it easily by employing techniques such as feature extractions. This paper also explains different datasets used to develop intrusion detection, and this paper chooses the 5G NIDD dataset as it is created by actual network data packets rather than simulations. Incremental learning is heavily researched and analyzed and expected to provide excellent results because the model is trained on new data, and there will be precise attack detection.

E. DTL-IDS: Deep Transfer Learning-based Intrusion Detection System in 5G Networks

This study discusses a comprehensive approach to address the challenges of intrusion detection in modern 5G networks. With the increasing complexity and dynamic nature of these networks, traditional Intrusion Detection Systems (IDS) face difficulties in keeping pace with evolving attack techniques. The authors propose leveraging Deep Transfer Learning (DTL)

to enhance IDS performance by transferring knowledge from a source domain [2], where labeled data is abundant, to a target domain, typically characterized by limited labeled data specific to 5G environments. The study systematically evaluates various deep learning algorithms as classifiers for detecting intrusions in 5G networks. Through experiments conducted on a carefully curated dataset collected from a 5G test-bed and the 5G-NIDD dataset, the authors demonstrate the effectiveness of the proposed DTL-based approach. They meticulously outline the pre-processing steps, base model construction, and transfer learning phase, providing insights into the methodology's implementation. Furthermore, the paper discusses the performance evaluation results, emphasizing metrics such as accuracy, recall, precision, and F1-score. The results indicate significant improvements in intrusion detection accuracy when employing DTL compared to traditional approaches. Through a thorough literature review, the study contextualizes the proposed methodology within the broader landscape of transfer learning-based intrusion detection solutions, highlighting its novelty and contributions to the field. Overall, the paper provides valuable insights and contributions to the domain of cyber-security in 5G networks, paving the way for more effective and adaptive intrusion detection systems.

F. Wireless Intrusion and Attack Detection for 5G Networks using Deep Learning Techniques

Researchers have explored various approaches to address this challenge, including the use of deep learning techniques for intrusion and attack detection. This study presents a deep learning-based intrusion detection system (IDS) for 5G wireless networks [1]. The researchers used an autoencoder with a deep neural network (DNN) algorithm to build the deep learning-based IDS, which was evaluated using the Aegean Wi-Fi Intrusion dataset (AWID). The AWID dataset contains data on various types of attacks, such as Flooding, Impersonation, and Injection. The deep learning-based IDS achieved an impressive accuracy of 99% in detecting the attack types in the AWID dataset. The results demonstrate that the deep learning-based approach can provide a robust security framework for 5G networks by achieving high levels of accuracy and precision in detecting intrusions and attacks.

III. Data Preprocessing

For the purpose of our research, we utilized the comma-separated value (CSV) file from the 5G Network Intrusion Detection Dataset (5G-NIDD) [5], which contains concatenated data encompassing all attack scenarios. This consolidated file provided a comprehensive representation of the network traffic data, including both malicious and benign activities. The dataset consists of 1,215,890 rows, with 477,737 representing benign activities and 738,153 representing various types of malicious attacks. Figure 1 shows the distribution of these benign and malicious activities present in the dataset.

To ensure the input dataset was suitable for the machine learning models considered in this study, we performed various preprocessing steps on the dataset.

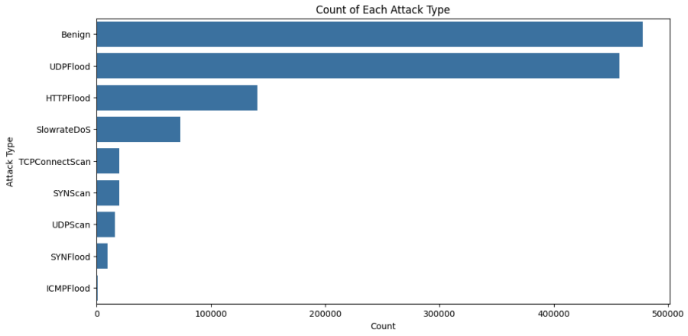

Fig. 1. Distribution of benign and various types of malicious attacks

A. Handling Missing Values

The dataset underwent an initial cleaning process, where duplicate entries were removed. Additionally, a few rows belonging to the benign class were dropped from the dataset due to a significant number of missing values across various columns. Similarly, certain columns (sVid and dVid) were removed as they were deemed extraneous to the analysis. Figure 2 depicts the degree of correlation between columns with missing values.

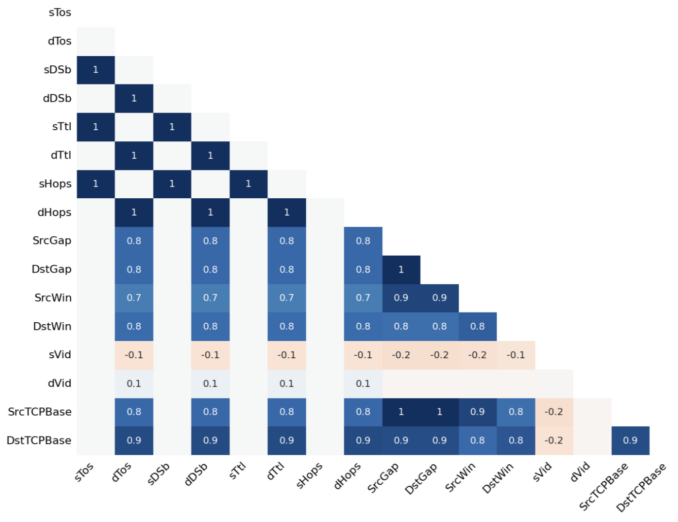

Fig. 2. Missing value analysis - The shade of blue represents the degree of correlation of missing values between columns

Missing values in each row were then imputed with the most frequent value (mode) within each group, as determined by the Attack Type column, using a custom function. The mode was selected as the imputation method because each of these columns had only 6-7 different values, indicating that the data was categorical or ordinal in nature. Utilizing the mode in this instance preserved the categorical nature of the data while avoiding the introduction of additional values that may not be representative of the original data distribution. For a subset of columns, an iterative imputer (MICE) was employed to handle any remaining missing data.

B. One Hot Encoding and Normalization

To get the dataset ready for analysis, we used one-hot encoding and normalization. Utilizing one-hot encoding made it easier to use machine learning methods by converting categorical variables into binary dummy variables. Every set of dummy variables has one category eliminated in order to prevent duplication.

Following encoding, the dataset was divided into target variables (y), which stood in for the 'Attack Type' column, and characteristics (X). Next, superfluous columns were removed from the feature set X and the encoded dummy variables were added.

We used custom normalization to ensure consistent scaling and to facilitate the convergence of machine learning algorithms. While categorical features (one-hot encoded dummy variables) were adjusted using a Min-Max Scaler function to ensure values varied between 0 and 1, numerical features were standardized using a Standard Scaler function.

C. Label Encoding, Mutual Information, and Principal Component Analysis

To convert the target variable 'Attack Type' categories into integers and prepare them for machine learning, we transformed the categories to integers using the LabelEncoder from Scikit-Learn [4].

Following this, we split the preprocessed data into train and test sets allocating 15% of the data for testing and 85% for training.

We devised a method for measuring the mutual information [7] between the target variable and characteristics. This metric takes into account both linear and non-linear correlations to determine how relevant each attribute is to the aim. It helps choose the most illuminating characteristics, which may improve interpretability, reduce overfitting, and increase model performance. Figure 3 shows the top 25 features identified.

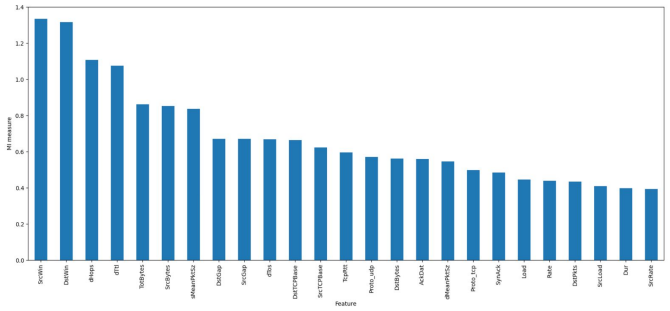

Fig. 3. Mutual Information Scores of Features

We implemented PCA, a technique for dimensionality reduction. It generates uncorrelated variables (principal components) to capture essential data patterns. PCA [7] reduces dimensionality while preserving information, crucial for high-dimensional data. It improves computational efficiency, reduces overfitting, and aids visualization. The function requires scaled data and the desired number of components, returning a represention of the data in reduced-dimensional space, enhancing model performance.

After selecting features and reducing dimensionality, we applied stratified k-fold cross-validation [4] to evaluate model performance. This maintains class proportions across folds, crucial for imbalanced data. Each fold served as both training and validation sets iteratively, ensuring a reliable performance estimate. Performance metrics (accuracy, precision, recall, and F1-score) were averaged across iterations to gauge overall model effectiveness.

IV. Machine Learning Models

A. Dense Neural Network with Autoencoder

In this subsection, we introduce a Dense Neural Network (DNN) architecture enhanced with an autoencoder component for feature extraction and dimensionality reduction. This model leverages the power of deep learning to learn complex patterns and representations from the input data while reducing its dimensionality, thereby improving computational efficiency and generalization performance.

1) Model Architecture: The Dense Neural Network (DNN) with autoencoder is designed to efficiently extract and represent essential features from the input data while reducing its dimensionality. Unlike traditional autoencoder architectures consisting of both encoder and decoder components, our model focuses solely on the encoder part for dimensionality reduction. Figure 4 shows the model architecture of the autoencoder and the neural network.

Layer (type)	Output Shape	Param #
Linear-1	[-1, 32]	2,016
Tanh-2	[-1, 32]	0
Linear-3	[-1, 16]	528
Tanh-4	[-1, 16]	0
Linear-5	[-1, 12]	204
Tanh-6	[-1, 12]	0
Linear-7	[-1, 32]	416
BatchNorm1d-8	[-1, 32]	64
ReLU-9	[-1, 32]	0
Dropout-10	[-1, 32]	0
Linear-11	[-1, 9]	297
Softmax-12	[-1, 9]	0

Fig. 4. Dense Neural Network model architecture

The primary component of the architecture is the encoder, responsible for extracting essential features from the input data. The encoder consists of a series of dense layers with Tanh activation function. Tanh introduces nonlinearity into the model, enabling it to learn complex mappings from the input to the encoded representation while ensuring that the encoded values are bounded between -1 and 1. The encoder output serves as input for the Dense Neural Network, featuring ReLU activation in its hidden layers. ReLU is well-suited for deep neural networks as it efficiently addresses the vanishing gradient problem and accelerates convergence during training. By incorporating ReLU activation functions into the DNN layers, we introduce nonlinearity and enable the model to capture complex patterns and relationships within the encoded data.

2) Hyperparameters and Training Setup: The model is trained using a mini-batch gradient descent approach with a batch size of 64. We employ the Adam optimizer, a widely used optimization algorithm known for its efficiency and effectiveness in training deep neural networks. The learning rate is set to 0.001, with a weighted decay of 10e-8 to promote smoother convergence.

To evaluate the model's performance and prevent overfitting, we employed a stratified train-validation-test split with a ratio of 70-15-15, ensuring that each class is proportionally represented in the training, validation, and test sets. This helps the model generalize well to unseen data.

We trained the model for 8 epochs, monitoring both the training and validation losses. Model checkpointing is employed to save the model parameters only when both the training and validation losses decrease, ensuring that we capture the best-performing model while avoiding overfitting.

3) Loss Function and Regularization: We employed cross-entropy loss for multi-class classification tasks, which effectively measures the disparity between predicted class probabilities and actual labels, aiding in accurate classification. Additionally, for binary classification, binary cross-entropy loss was utilized.

To prevent overfitting and improve the model's generalization performance, we incorporated batch normalization and dropout regularization techniques. Batch normalization normalizes the activations of each layer, reducing internal covariate shift and accelerating the training process. Dropout randomly disables a fraction of neurons during training, forcing the model to learn robust features and reducing the risk of overfitting.

B. K-Nearest Neighbors

In this subsection, we introduce K-Nearest Neighbors classifier(KNN) [4] architecture enhanced with an PCA for feature extraction and mutual information for feature selection. This model leverages the power of proximity to classify about the grouping of an individual data point.

1) Model Architecture: As the current dataset had 60+ features, classifying that directly will invoke huge computational overload on the system. So before classifying it, we need to do several dimensionality reduction techniques to improve the accuracy.

2) Fitting training model: Once we get the 15 features which are extracted from the 20 features we received from the mutual information, we are fitting the training dataset with the KNN classifier using Stratified K-fold such that the training and validation sample differ in each fold. We are using 5-folds such that all folds/subset of training dataset will be a validation/testing dataset. Once the training is completed, we are predicting the testing dataset with the KNN classifier.

C. Support Vector Machine (SVM) Model

The Support Vector Machine (SVM) is a powerful supervised learning algorithm, primarily employed for classification and regression problems. Its primary objective is to identify the optimal hyperplane that effectively separates distinct classes of data within the feature space. In our research, we implement the SVM model using the scikit-learn [4] Python library, which is well-suited for such machine learning applications. The top 25 features of our dataset were selected based on the mutual information technique, ensuring that the most relevant features are utilized for model training. To further enhance the model's performance by reducing dimensionality while retaining essential information, PCA was employed, keeping only the top 15 principal components.

For our SVM classifier, the Radial Basis Function (RBF) kernel was chosen due to its effectiveness in handling non-linear data separations. We divided the dataset into training and testing sets, comprising 85% and 15% of the data respectively. The training process was rigorously conducted using the k-fold cross-validation technique, which helps in validating the stability and reliability of our model by training it on multiple subsets of the data. This method not only contributes to a more stable and reliable model but also facilitates performance assessment across diverse data segments. Upon completion of the training phase, the model's predictive capabilities were thoroughly evaluated on the holdout testing set.

D. Random Forest Classification Model

Random Forest is a strong ensemble learning method that combines multiple decision trees. We implemented this method to help us prevent overfitting and ensure reliable classification results.

We evaluated each feature's importance to understand data patterns. This guided our feature selection process to focus on the most important variables for accurate classification. Random Forest works well with datasets having many features, which is very much suitable for our case. Its parallel processing capability allowed efficient handling of our large, complex dataset. This scalability ensures practicality regardless of dataset size.

We tuned hyperparameters to optimize the Random Forest model's performance. We disabled bootstrap sampling by setting 'bootstrap' to False so each tree was trained on the full dataset. We used the entropy criterion to measure split quality in the trees. We allowed unlimited tree depth until the leaves were pure. We set the minimum leaf samples and minimum split samples to 2. Finally, we used 50 trees to balance model complexity and efficiency.

In conclusion, our Random Forest implementation demonstrates its robustness, versatility, interpretability, scalability, and tunability. These aspects make it well-suited for our classification problem to provide accurate, reliable results tailored to our data.

ANALYSIS AND DISCUSSION

The computational experiments conducted on the Sol supercomputer at Arizona State University utilized NVIDIA A100 GPUs, with each node featuring 15 GB of memory. This powerful computational infrastructure enabled the rigorous evaluation of our machine-learning models under optimal conditions,

TABLE I
RESULTS OF DIFFERENT EVALUATION METRICS OBTAINED FOR BINARY CLASSIFICATION

Model	Precision	Recall	F1-Score	Accuracy	Training Time (s)	Prediction Time (s)
Dense Neural Network	1.0	1.0	1.0	1.0	40.37602400779724	903.8087875843048
KNN	0.9997	0.9997	0.9997	0.9997	61.050031661987305	8.113478899002075
Random Forest	0.9998	0.9998	0.9998	0.9998	58.929343461990356	7.744907379150391
SVM	0.9999	0.9999	0.9999	0.9999	32354.052695035934	12.178013563156128

TABLE II
RESULTS OF DIFFERENT EVALUATION METRICS OBTAINED FOR MULTICLASS CLASSIFICATION

Model	Precision	Recall	F1-Score	Accuracy	Training Time (s)	Prediction Time (s)
Dense Neural Network	0.9971	0.9970	0.9970	0.9970	1676.5337464809418	41.186461448669434
KNN	0.9923	0.9923	0.9923	0.9923	2066.196816921234	360.3606536388397
Random Forest	0.9963	0.9963	0.9963	0.9963	521.323098897934	0.5431344509124756
SVM	0.9925	0.9923	0.9922	0.9922	13740.027164697647	348.37729501724243

TABLE III
RESULTS OF DIFFERENT EVALUATION METRICS OBTAINED FOR MULTICLASS CLASSIFICATION WITH CLASSES

Model	Attack Type	Precision	Recall	F1-Score	Accuracy	Training Time (s)	Prediction Time (s)
Dense Neural Network	Benign	0.9997	0.9999	0.9998	0.9970	1676.533	41.186
	HTTP Flood	0.9909	0.9957	0.9933			
	ICMP Flood	0.9885	1.00000	0.9942			
	SYN Flood	0.9928	0.8552	0.9189			
	SYN Scan	0.9966	0.9973	0.9970			
	Slowrate DoS	0.99667	0.99733	0.99700			
	TCP Connect Scan	0.93626	0.99634	0.96537			
	UDP Flood	1.00000	1.00000	1.00000			
	UDP Scan	0.99706	0.99664	0.99685			
Random Forest	Benign	1.00000	1.00000	1.00000	0.9963	521.323	0.543
	HTTP Flood	0.98157	0.98845	0.98500			
	ICMP Flood	1.00000	1.00000	1.00000			
	SYN Flood	0.99658	0.99863	0.99760			
	SYN Scan	0.99933	0.99601	0.99767			
	Slowrate DoS	0.97736	0.96435	0.97081			
	TCP Connect Scan	0.99801	0.99867	0.99834			
	UDP Flood	1.00000	1.00000	1.00000			
	UDP Scan	0.99707	0.99832	0.99770			
KNN	Benign	1.00000	0.99941	0.99971	0.9923	2066.196	360.360
	HTTP Flood	0.96652	0.97041	0.96846			
	ICMP Flood	1.00000	1.00000	1.00000			
	SYN Flood	0.99656	0.99246	0.99450			
	SYN Scan	0.99403	0.99667	0.99535			
	Slowrate DoS	0.93403	0.93582	0.93492			
	TCP Connect Scan	0.98946	0.99867	0.99404			
	UDP Flood	1.00000	1.00000	1.00000			
	UDP Scan	0.99958	0.99790	0.99874			
SVM	Benign	0.99993	0.99999	0.99996	0.9922	13740.027	348.377
	HTTP Flood	0.95193	0.99470	0.97284			
	ICMP Flood	0.96648	1.00000	0.98295			
	SYN Flood	0.99603	0.86145	0.92387			
	SYN Scan	0.93516	0.99800	0.96556			
	Slowrate DoS	0.99980	0.90127	0.94798			
	TCP Connect Scan	0.99371	0.99867	0.99619			
	UDP Flood	1.00000	1.00000	1.00000			
	UDP Scan	1.00000	0.99832	0.99916			

ensuring robust training and testing processes. Performance metrics were meticulously assessed to gauge the efficacy of the developed models in detecting diverse cyber threats within the 5G network landscape. These metrics, including accuracy, precision, recall, and F1 score, offer valuable insights into the models' capabilities and limitations.

Of particular significance is the focus on minimizing false negatives, as these instances represent missed detections of potential threats, posing a significant risk to network security. By reducing false negatives, our models aim to enhance overall threat detection accuracy and mitigate potential vulnerabilities in the 5G ecosystem. Additionally, it is essential to examine misclassification patterns and their underlying causes. Understanding why certain classes are prone to misclassification can provide valuable insights for refining model architectures and optimizing feature selection processes. By addressing these underlying causes, we can enhance the models' ability to accurately classify diverse cyber threats in real-world scenarios.

Table I presents the evaluation metrics for binary classification, highlighting the models' performance in distinguishing between threat and non-threat instances. Table II and Table III showcases the evaluation metrics for multiclass classification, illustrating the models' effectiveness in identifying specific types of cyber threats across different categories. Through a comprehensive analysis of these metrics and a thorough investigation into misclassification patterns, our research aims to contribute to the development of more robust and effective machine learning solutions for enhancing the security posture of 5G networks.

Most of the misclassifications occurred between specific pairs of attack types, each exhibiting distinct characteristics. For instance, the HTTPFlood attack mimics human behavior by generating a large volume of seemingly legitimate requests, which can lead to confusion with other attack types. Conversely, in SlowRateDOS attacks, malicious activities occur at a slower pace, potentially causing misclassification due to their subtle nature. Additionally, discrepancies arise between SYNFlood and TCPConnectScan attacks, particularly in their request patterns. In SYNFlood attacks, the third request (ACK bit) is typically skipped, whereas it is sent in TCPConnectScan. Consequently, the first two requests in both attack types may exhibit similar characteristics, leading to misclassification. These nuances highlight the challenges inherent in accurately distinguishing between attack types and underscore the need for further refinement in feature extraction and model architecture to enhance classification accuracy in 5G network security systems.

Discussion About Potential Future Work

As cyber threats keep evolving, it is important to keep improving and adapting intrusion detection models to combat new types of attacks. One area for future research is extending existing models to handle a wider range of attack scenarios. This includes investigating advanced techniques to make models more robust against sophisticated evasion and adversarial attacks. Incorporating methods to detect behavioural anomalies can also help identify insider threats and new zero-day exploits that signature-based detection may miss. Additionally, with more Internet of Things (IoT) devices being used, it is crucial to extend intrusion detection to cover IoT threat detection. This involves developing specialized techniques for securing resource-limited IoT environments and diverse communication protocols, strengthening overall network security.

Deploying intrusion detection within 5G networks for real-time threat detection is an opportunity to enhance network security. One approach is integrating lightweight intrusion detection at the network edge, using edge computing for decentralized threat detection and mitigation. Integrating intrusion detection with Software-Defined Networking controllers allows dynamic network reconfiguration based on real-time threat intelligence. Using containerization makes deployment easier by running intrusion detection as microservices in containers. Hybrid cloud models provide scalability and flexibility, using cloud resources for processing while keeping sensitive data on-premises. Integrating with threat intelligence platforms and continuous monitoring/retraining ensures intrusion detection stays effective against evolving threats over time, improving overall network security posture.

Conclusions

In conclusion, this research underscores the superiority of machine learning (ML) over traditional approaches in intrusion detection and classification within network security. By evaluating ML models such as Deep Neural Networks (DNN) with Autoencoders, K-Nearest Neighbors (KNN), Support Vector Machines (SVM), and Random Forest Classifier, our study consistently demonstrated their exceptional performance. With detection rates nearing 100% and classification accuracy around 99%, our models exhibited remarkable efficiency in identifying and categorizing intrusions, critically minimizing false negatives which can leave systems vulnerable to undetected threats.

These findings highlight the transformative potential of ML-based intrusion detection systems to fortify network security defences. By leveraging ML's capabilities, organizations can proactively detect and mitigate a wide array of cyber threats while significantly reducing the risk of intrusions going undetected. However, ongoing efforts are needed to refine ML algorithms, address challenges such as adversarial attacks, and ensure the adaptability of models to evolving threat landscapes. Ultimately, embracing ML holds promise for ushering in a new era of resilient and proactive cybersecurity measures that minimize false negatives in the face of escalating cyber threats.

Acknowledgment

We would like to express our gratitude to Arizona State University for the support and resources provided throughout the duration of this research. Specifically, we acknowledge the infrastructure and computing resources made available by the institution, which were instrumental in training and validating the models presented in this paper. We are deeply grateful for

their commitment to advancing research and innovation in the field of Computer Science, Machine Learning and Artificial Intelligence

REFERENCES

[1] Bayana Alenazi and Hala Eldaw Idris. Wireless intrusion and attack detection for 5g networks using deep learning techniques. *International Journal of Advanced Computer Science and Applications*, 12(7), 2021.

[2] Behnam Farzaneh, Nashid Shahriar, Abu Hena Al Muktadir, and Md Shamim Towhid. Dtl-ids: Deep transfer learning-based intrusion detection system in 5g networks. In *2023 19th International Conference on Network and Service Management (CNSM)*, pages 1–5. IEEE, 2023.

[3] Cheolhee Park, Kyungmin Park, Jihyeon Song, and Jonghyun Kim. Distributed learning-based intrusion detection in 5g and beyond networks. In *2023 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, pages 490–495. IEEE, 2023.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[5] Sehan Samarakoon, Yushan Siriwardhana, Pawani Porambage, Madhusanka Liyanage, Sang-Yoon Chang, Jinoh Kim, Jonghyun Kim, and Mika Ylianttila. 5g-nidd: A comprehensive network intrusion detection dataset generated over 5g wireless network, 2022.

[6] Sehan Samarakoon, Yushan Siriwardhana, Pawani Porambage, Madhusanka Liyanage, Sang-Yoon Chang, Jinoh Kim, Jonghyun Kim, and Mika Ylianttila. 5g-nidd: A comprehensive network intrusion detection dataset generated over 5g wireless network. *arXiv preprint arXiv:2212.01298*, 2022.

[7] Zihao Wang, Kar Wai Fok, and Vrizlynn L. L. Thing. Exploring emerging trends in 5g malicious traffic analysis and incremental learning intrusion detection strategies, 2024.