



MACHINE LEARNING-BASED ANOMALY DETECTION IN POST-QUANTUM TLS 1.3 TRAFFIC: A COMPARATIVE ANALYSIS ON THE CIC-PQC_OAV V1 DATASET

Serkan KESKİN^{id}

Burdur Mehmet Akif Ersoy Üniversitesi, Teknik Bilimler Meslek Yüksekokulu, Veri Tabanı, Ağ
Tasarımı Ve Yönetimi Bölümü, Burdur, Türkiye

Corresponding Author: skeskin@mehmetakif.edu.tr

ÖZET: Kuantum sonrası kriptografik algoritmaların TLS 1.3 protokolüne entegrasyonu, ağ trafiğinin davranışsal özelliklerini değiştirebilir ve mevcut anomali tespit sistemlerinin performansını etkileyebilir. TLS'nin önceki sürümlerinde güvenli iletişim büyük ölçüde klasik anahtar değişim mekanizmalarına dayanırken, TLS 1.3 oturumlarına kuantum sonrası algoritmaların dahil edilmesi farklı trafik davranışlarına yol açabilir. Bu değişikliği göz önünde bulundurarak, bu çalışmada CIC-PQC_OAV v1 veri kümesi üzerinde makine öğrenimi tabanlı anomali tespiti gerçekleştirilmiştir. Karar Ağacı, K-En Yakın Komşu, Destek Vektör Makinesi, Rastgele Orman ve XGBoost algoritmaları, aynı eğitim, doğrulama ve test bölümleri kullanılarak uygulanmıştır. Modeller, Doğruluk, Keskinlik, Geri Çağırma, F1 Skoru ve ROC-AUC açısından karşılaştırılmıştır. Deney sonuçları, Karar Ağacı ve Rastgele Orman algoritmalarının %97,79 Doğruluk ve %90,31 F1-Skoru ile en dengeli performansı sergilediğini göstermektedir. XGBoost ise sınıf ayrıştırılabilirliği açısından öne çıkmış ve %98,35'lik bir ROC-AUC değeri elde etmiştir. Bu bulgular, CIC-PQC_OAV v1 veri seti üzerinde geleneksel makine öğrenimi modellerinin performans özelliklerini ortaya koymaktadır. Sonuçlar, kuantum sonrası TLS 1.3 ortamlarında anomali tespiti için ampirik bir referans noktası sağlar ve farklı ağ trafiği ayarlarında karşılaştırmalı değerlendirmeler içeren gelecekteki çalışmaları destekleyebilir.

Anahtar Kelimeler: Post-kuantum kriptografi, TLS 1.3, CIC-PQC_OAV v1, anomali tespiti, makine öğrenmesi

ABSTRACT: The integration of post-quantum cryptographic algorithms into the TLS 1.3 protocol may alter the behavioral characteristics of network traffic and affect the performance of existing anomaly detection systems. While secure communication in earlier uses of TLS largely relied on classical key exchange mechanisms, the inclusion of post-quantum algorithms in TLS 1.3 sessions may lead to different traffic behaviors. Considering this change, this study performs machine learning-based anomaly detection on the CIC-PQC_OAV v1 dataset. Decision Tree, K-Nearest Neighbor, Support Vector Machine, Random Forest, and XGBoost algorithms were applied using the same training, validation, and test partitions. The models were compared in terms of Accuracy, Precision, Recall, F1-Score, and ROC-AUC. The experimental results show that Decision Tree and Random Forest achieved the most balanced performance, with an Accuracy of 97.79% and an F1-Score of 90.31%. XGBoost stood out in terms of class separability, achieving a ROC-AUC value of 98.35%. The findings demonstrate the performance characteristics of traditional machine learning models on the CIC-PQC_OAV v1 dataset. The results

provide an empirical baseline for anomaly detection in post-quantum TLS 1.3 environments and may support future studies involving comparative evaluations across different network traffic settings.

Keywords: Post-quantum cryptography, TLS 1.3, CIC-PQC_OAV v1, anomaly detection, machine learning

1. INTRODUCTION

At present, a large proportion of communication over the Internet is carried out through encrypted protocols. In critical applications such as e-commerce, cloud computing, online banking, and public services, strong cryptographic mechanisms are required to ensure data security [1]. In this context, the Transport Layer Security (TLS) protocol is one of the fundamental security technologies that protects the confidentiality and integrity of communication between clients and servers. TLS 1.3 is widely used in contemporary Internet infrastructures because it provides lower latency, a simplified handshake mechanism, and stronger security properties compared with previous versions [2].

However, advances in quantum computing have raised important discussions regarding the future security of existing cryptographic systems. In particular, the possibility that public-key cryptographic algorithms may be broken by quantum computers in the future has led researchers to develop quantum-resistant alternatives. Post-Quantum Cryptography (PQC) refers to next-generation cryptographic approaches designed to provide security against both classical and quantum computers. The standardization of the ML-KEM algorithm through the FIPS 203 standard published by NIST in 2024 is considered an important milestone in the transition toward PQC [3], [4].

The widespread adoption of post-quantum cryptographic algorithms affects not only cryptographic infrastructures but also the monitoring and intrusion detection systems used in network security [5]. Next-generation key exchange and signature mechanisms may change network characteristics such as packet sizes, session establishment times, traffic volume, and connection behavior [6]. These changes may directly influence the performance of intrusion detection systems trained on conventional network traffic. Therefore, during the transition to PQC, not only secure communication mechanisms but also the security solutions that monitor them should be reassessed. Roadmaps published by the European Union and several countries also emphasize that the PQC transition represents a strategic transformation affecting the entire cybersecurity ecosystem [7].

These emerging requirements in network security have increased interest in machine learning-based anomaly detection methods. Although traditional signature-based systems can successfully detect known threats, they may be insufficient for identifying changing traffic behavior and next-generation attacks [8]. Machine learning methods are widely used in anomaly detection studies because they can learn complex patterns in network traffic and automatically detect unusual behavior.

When existing studies are examined, machine learning-based intrusion and anomaly detection approaches are seen to be mostly evaluated on classical network datasets such as CICIDS2017, CICIDS2018, UNSW-NB15, and similar benchmarks [9]. Although these datasets provide an important basis for network security research, they do not directly represent the traffic characteristics that emerge when post-quantum cryptographic algorithms are integrated into TLS

1.3 sessions. Post-quantum key exchange mechanisms may produce different behavior in variables such as TLS record lengths, connection duration, packet volume, and entropy. Therefore, it should not be assumed that an anomaly detection model that performs successfully on conventional network traffic will necessarily show the same level of success on post-quantum TLS traffic [10], [11].

In order to solve the above problem, the current study examines the anomaly detection problem in the post-quantum TLS 1.3 traffic through CIC-PQC_OAV v1 dataset. The Decision Tree (DT), K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Random Forest (RF), and XGBoost methods have been tested by using the same training, validation, and testing datasets and their performances have been compared through Accuracy, Precision, Recall, F1-Score, and ROC-AUC metrics. Thus, both overall classification accuracy and anomaly class classification performance have been taken into account at the same time.

The major contributions of this study can be listed as follows:

- Comparison of various machine learning algorithms under similar experimental settings for CIC-PQC_OAV v1 dataset.
- Examination of the anomaly detection problem in the post-quantum TLS 1.3 traffic in a different context than classical IDS datasets.
- Evaluation of model performance in terms of Accuracy as well as Precision, Recall, F1-Score, and ROC-AUC measures at the same time.
- The results of the study suggest that anomaly detection systems should be re-evaluated in the PQC era.

2. LITERATURE REVIEW

Recent studies in network anomaly detection indicate that machine learning and deep learning methods can achieve high success rates. In particular, studies conducted on widely used datasets such as CICIDS2017, CICIDS2018, and UNSW-NB15 have reported accuracy values above 95%.

Adewole et al. (2025) achieved an accuracy of 99.91% using the XGBoost algorithm on the CICIDS2017 dataset. In the same study, experiments conducted on the CICIOT2023 dataset reached an accuracy of 98.54%. These results demonstrate that the XGBoost algorithm can achieve high performance in different network environments [12].

Talukder et al. (2024) evaluated machine learning-based intrusion detection methods on large and imbalanced datasets. The authors reported an accuracy of 99.95% with the Extra Trees algorithm and 99.59% with the RF algorithm on the UNSW-NB15 dataset. In the same study, accuracy values of 99.99% on CICIDS2017 and 99.94% on CICIDS2018 were reported. These results indicate that ensemble learning-based methods are highly successful in network intrusion detection [13].

More et al. (2024) achieved an accuracy of 99.40% with the RF algorithm on the UNSW-NB15 dataset. After the application of feature selection methods, the accuracy increased to 99.45%. This suggests that appropriate feature selection can contribute positively to model performance [14].

Deep learning-based methods have also produced notable results in recent years. Jablaoui et al. (2025) achieved an accuracy of 97.95% using a CNN-GRU-based hybrid model on the NF-CSE-CIC-IDS2018 dataset. The same model reached an accuracy of 98.60% on the NF-UNSW-NB15 dataset. These results show that deep learning methods can adapt to different types of network traffic [15].

Alsulami et al. (2024) proposed an intrusion detection model based on a multi-attention mechanism and obtained an accuracy of 98.82% on the UNSW-NB15 dataset and 99.12% on the CICIDS2017 dataset. The authors stated that attention mechanisms improve classification performance by learning important patterns in network traffic more effectively [16].

Overall, the results reported in the literature show that RF, Extra Trees, XGBoost, and deep learning-based hybrid models can reach high accuracy values in network intrusion detection. However, most of these studies have been conducted on traditional network datasets such as CICIDS2017, CICIDS2018, and UNSW-NB15. Studies on datasets containing post-quantum cryptographic TLS 1.3 traffic remain highly limited. Therefore, studies conducted on the CIC-PQC_OAV v1 dataset are expected to contribute to the evaluation of anomaly detection systems in post-quantum network environments.

3. MATERIALS AND METHODS

3.1. Dataset

This study uses the CIC-PQC_OAV v1 dataset developed by the Canadian Institute for Cybersecurity (CIC). The dataset was created to analyze network traffic generated by the integration of post-quantum cryptographic algorithms into the TLS 1.3 protocol and to evaluate anomaly detection systems. The CIC-PQC_OAV v1 dataset is publicly available through the Canadian Institute for Cybersecurity (CIC), enabling researchers to reproduce the experimental setup and perform comparative analyses in post-quantum network security studies [17].

The CIC-PQC_OAV v1 dataset includes network flows obtained from post-quantum TLS sessions generated using ML-KEM-768, ML-KEM-1024, and X25519-based key exchange mechanisms, in addition to classical TLS 1.3 traffic. The dataset contains various features related to traffic volume, connection duration, TLS record lengths, entropy values, and cryptographic context [17].

The dataset used in this study consists of 40,010 records and 35 features. The target variable, label, takes the value “0” for normal traffic and “1” for anomalous traffic. The feature groups included in the dataset are summarized in Table 2.

Table 1 presents the features in the CIC-PQC_OAV v1 dataset under simplified basic groups. Although the dataset contains 36 columns in total, the label column was used as the target variable. The split, taxonomy, and ID fields were not included in model training because they represent data partitioning, anomaly type information, and record identifiers, respectively. Accordingly, 32 independent features were used to train the machine learning models.

An examination of the table shows that the dataset is not limited to classical network flow features. In addition to traffic volume, TLS record lengths, entropy values, and timing information, it includes features representing post-quantum key exchange algorithms such as ML-KEM-768, ML-KEM-1024, and X25519. This indicates that the dataset provides a suitable structure for analyzing the behavioral characteristics of post-quantum TLS traffic.

Table 1. Feature groups in the CIC-PQC_OAV v1 dataset

Feature Group	Number of Features	Representative Features
Traffic volume and TLS record features	4	e2c_total_bytes, e2_client_size, e2_client_record_len
Entropy and timing features	4	e4_entropy_h, e5_entropy_c, e6b_flow_duration_ms
Certificate and signature features	3	e3_cert_parsed, e3_sig_alg_Unknown(0x0028)
Key exchange algorithm features	6	e1_alg_suite_mlkem768, e1_alg_suite_mlkem1024, e1_alg_suite_x25519
Protocol and connection outcome features	5	e8_proto_context_TLSv1.3, e9_conn_outcome_Success
TLS version and cipher suite features	10	e1b_tls_version_TLS1.0, e1b_ciphersuite_54, e2b_ciphersuite_60
Target variable	1	label
Fields excluded from training	3	split, taxonomy, ID

In addition, protocol context, connection outcome, TLS version information, and cipher suite features allow not only volumetric but also protocol-level behavior of network sessions to be reflected in the model. Therefore, unlike traditional IDS datasets, the CIC-PQC_OAV v1 dataset constitutes a more distinctive resource for evaluating anomaly detection performance in post-quantum cryptographic network environments.

3.2. Data Preprocessing

To enable machine learning models to produce reliable and comparable results, several preprocessing steps were applied to the dataset. First, the variables in the dataset were examined, and the columns to be used in the modeling process were determined. In this context, the label column was selected as the target variable, while the ID, split, and taxonomy columns were not used as model inputs.

The ID column was excluded from model training because it only serves as a record identifier. The taxonomy column was not used as a model input because it contains anomaly type information. Including this column in the model could cause data leakage in the classification process; therefore, it was excluded. The split column was also not used as an independent variable; instead, it was used only to determine the predefined training, validation, and test partitions. As a result, 32 independent features were used in model training.

Missing value analysis revealed missing values in four features: e5_entropy_c, e6_time_char, e2_client_size, and e2_server_record_len. To prevent missing values from adversely affecting model training, missing observations in numerical variables were imputed using median values calculated from the training data. The same transformation was then applied to the validation and test sets, thereby preventing information from the test data from leaking into the training process.

The dataset contains both numerical features and binary-coded logical variables. Binary-coded variables were converted into 0 and 1 values to make them suitable for model training.

StandardScaler was applied to numerical variables to prevent scale differences from adversely affecting model performance. In this process, each numerical feature was transformed to have zero mean and unit standard deviation. Standardization is particularly important for KNN and SVM, as these algorithms are sensitive to feature scale differences.

The predefined split column in the dataset was used to create the training, validation, and test sets. Accordingly, 24,006 records were assigned to the training set, 8,002 records to the validation set, and 8,002 records to the test set. The training set was used for model learning, the validation set for model adjustment, and the test set for calculating final performance results. The natural class distribution of the dataset was preserved, and no resampling method was applied. Thus, the study aimed to evaluate the natural anomaly detection performance of the algorithms on post-quantum TLS 1.3 traffic.

The original dataset contains 36 columns. After excluding the label, split, taxonomy, and ID fields, a total of 32 features were retained for model development.

Within the CIC-PQC_OAV v1 dataset, the anomaly class represents deviations from expected TLS 1.3 communication behavior. These deviations include irregularities associated with post-quantum TLS sessions, abnormal communication patterns, and protocol-level events identified during the dataset generation process.

Table 2. Distribution of Samples Across Dataset Partitions

Dataset Partition	Normal	Anomaly	Total
Training	21,000	3,006	24,006
Validation	7,000	1,002	8,002
Test	7,000	1,002	8,002
Total	35,000	5,010	40,010

As presented in Table 2, the CIC-PQC_OAV v1 dataset exhibits a moderate class imbalance across all predefined partitions. The original train, validation, and test splits were preserved throughout the study to prevent data leakage and ensure consistency in the comparative evaluation of the machine learning models.

Table 3 summarizes the main preprocessing operations applied in this study. In this process, the target variable was first determined, columns not used in model training were removed, and modeling was performed with 32 independent features. Median imputation was applied to missing values to avoid data loss and to ensure that the models could operate on all records. In addition, scaling numerical features contributed to more balanced results, particularly for scale-sensitive algorithms such as KNN and SVM. The use of the predefined training, validation, and test partitions enabled the models to be objectively evaluated on previously unseen data.

Table 3. Operations applied in the data preprocessing process

Operation	Description
Determination of the target variable	The label column was used as the target variable.
Columns excluded from training	The ID, split, and taxonomy columns were not used as model inputs.
Number of independent features used	A total of 32 independent features were used in model training.
Missing value handling	Missing values were imputed using median values calculated from the training data.

Logical variable transformation	Boolean variables were encoded as 0 and 1.
Scaling	StandardScaler was applied to numerical features.
Data partitioning	The predefined train, val, and test partitions in the dataset were used.
Resampling	The class distribution was preserved, and no method such as SMOTE was applied.

3.3. Machine Learning Algorithms Used

In this study, five machine learning algorithms representing different learning approaches were used for anomaly detection in post-quantum TLS 1.3 network traffic. The algorithms were selected based on their common use in the network intrusion detection literature, their distinct classification mechanisms, and their suitability for comparative evaluation. Accordingly, DT, KNN, SVM, RF, and XGBoost algorithms were used.

3.3.1. Decision Tree

DT is a supervised machine learning algorithm which does classification through data splits using certain decision criteria. Due to the fact that relationships between the features are presented in the form of a tree, it makes the model very interpretable. Such feature gives an advantage in intrusion detection systems since understanding of the decision making process is important [18]. At the same time, single DT can be susceptible to overfitting in certain cases. That is why, DT algorithm was chosen for this research as a benchmark.

3.3.2. K-Nearest Neighbor

KNN is an example of a distance approach whereby classification of a data point is based on class details of its nearest neighbors. Despite its simplicity, KNN has been proven to be efficient in situations where there is similarity of patterns [19]. But since the performance of the algorithm depends on the scaling of features, numerical data was scaled before KNN application in this study.

3.3.3. Support Vector Machine

SVM technique is one of the most efficient classification algorithms which try to find the best separating hyperplane between different classes. SVM may give us good results particularly when there are some obvious separation boundaries among the classes. The reason behind using the SVM in this research is due to its efficiency in recognizing the normal and abnormal classes in network traffic data [20].

3.3.4. Random Forest

RF is an ensemble learning based classification technique which employs a number of DT together. As each DT gets trained on different sample sets and different feature sets, the generalization ability of the model improves and the possibility of overfitting reduces [21]. As RF has gained highly accurate results in network intrusion detection researches, this model was chosen for comparison in this research [22].

3.3.5. XGBoost

XGBoost is a state-of-the-art algorithm which works according to the principle of gradient boosting algorithm. It reduces the error rate by training weak learners iteratively. Due to its nature of regularization, high accuracy, and efficiency in handling big data, it is greatly preferred in most of the recent research related to network security and intrusion detection [23]. The reason for using the XGBoost model in this research is that of the possibility of obtaining high performance on post-quantum TLS traffic.

3.4. Experimental Environment and Implementation Process

The experiment was done using the Python programming language. The most common packages that are used in machine learning research, which include pandas, NumPy, scikit-learn, and XGBoost were utilized while performing data analysis, preprocessing, and modeling. The reading of the dataset, management of missing values, separation of variables, and feature scaling was done through the use of Python.

In the experimental process, the CIC-PQC_OAV v1 dataset was first imported into the working environment and its column structure was examined. The label column was then defined as the target variable, while the ID, split, and taxonomy columns were not used as model inputs. After this step, 32 independent features were obtained for model training.

The predefined split column in the dataset was used to construct the training, validation, and test sets. Therefore, the number of data included in the training set is 24,006, 8,002 in the validation set, and 8,002 in the test set. The training set data were used while the models were learning, the validation set data were used while tuning the models, and finally, the test set data were used to get the performance measures.

The DT, KNN, SVM, RF, and XGBoost algorithms were all trained and tested on the same dataset splits. Consequently, the algorithms' performances were compared under identical conditions. The accuracy, precision, recall, F1-score, and ROC-AUC of each model were measured to compare their performances.

The experimental process consisted of the following main steps:

1. Importing the dataset into the Python environment
2. Determining the target variable and independent features
3. Removing columns not used in model training
4. Imputing missing values with median values
5. Scaling numerical features using StandardScaler
6. Creating training, validation, and test sets
7. Training the machine learning algorithms
8. Evaluating the models on the test data
9. Comparing performance metrics

Through this process, the performance of different machine learning algorithms for anomaly detection in post-quantum TLS 1.3 traffic was systematically and comparably analyzed.

Moreover, all the tests have been performed in a computing environment based on Windows operating system with Intel CPU and 16 GB RAM. Python 3.x language has been used in combination with Pandas, NumPy, Scikit-learn, and XGBoost libraries. To facilitate

reproducibility of results, a constant random seed (random_state = 42) has been set wherever possible.

Further to this, the validation partition that was predefined in the CIC-PQC_OAV v1 dataset was utilized in the process of creating the model. In the case of classifiers that needed probabilistic results, the decision threshold used was 0.5, but ROC-AUC scores were calculated regardless of the decision threshold.

3.5. Hyperparameter Configuration

The hyperparameters for the different machine learning models that have been analyzed were chosen according to commonly used settings in the existing literature and from preliminary experiments. In order to maintain fairness, the parameter settings were not changed throughout all the experiments. The hyperparameters for each model are listed in Table 4.

Table 4. Hyperparameter Values Used

Model	Hyperparameters
DT	criterion=gini, random_state=42
KNN	n_neighbors=5, metric=euclidean
SVM	kernel=rbf, C=1.0
RF	n_estimators=100, random_state=42
XGBoost	n_estimators=100, learning_rate=0.1, max_depth=6

As shown in Table 4 below, the most frequently used settings of the hyperparameters have been chosen for all models analyzed in this paper. The reason for choosing these specific settings was to achieve experimental consistency.

4. RESULTS AND DISCUSSION

This section comparatively evaluates the anomaly detection performance of the machine learning algorithms applied to the CIC-PQC_OAV v1 dataset. DT, KNN, SVM, RF, and XGBoost algorithms were used in the study. The performance of the models was analyzed using Accuracy, Precision, Recall, F1-Score, and ROC-AUC metrics.

In the experimental evaluation, the predefined training, validation, and test partitioning provided in the dataset was preserved. The training set was used for model learning, the validation set for decision threshold and model evaluation processes, and the test set for obtaining the final performance results. In addition, missing value indicators were used to enable the models to produce more balanced results, and the decision threshold was determined on the validation set to maximize the F1-Score.

Table 5 shows that DT and RF achieved the most successful results in terms of Accuracy and F1-Score. Both algorithms reached an Accuracy of 97.79% and an F1-Score of 90.31%. This indicates that tree-based methods can effectively learn the traffic, TLS, and cryptographic context features in the CIC-PQC_OAV v1 dataset.

Table 5. Performance results of machine learning algorithms on the test data

Algorithm	Accuracy	Precision	Recall	F1-Score	ROC-AUC
-----------	----------	-----------	--------	----------	---------

Decision Tree	97.79	100.00	82.34	90.31	98.31
Random Forest	97.79	100.00	82.34	90.31	98.33
SVM	97.78	100.00	82.24	90.25	92.64
XGBoost	97.76	99.76	82.34	90.21	98.35
KNN	97.63	100.00	81.04	89.53	94.85

The Precision value of 100% for the RF and DT algorithms indicates that all records classified as anomalous by these models were truly anomalous. In other words, these models did not produce false positives on the test set. This is an important advantage in network security because a low false positive rate reduces the unnecessary alarm burden in security systems.

However, the Recall percentages of these models did not change from 82.34%. This means that there is a large percentage of the true anomaly cases that were detected correctly, but at the same time, there are anomalies that have been wrongly labeled as normal cases. This is very important in anomaly detection research.

XGBoost produced results very close to DT and RF in terms of Accuracy and F1-Score. The XGBoost model achieved an Accuracy of 97.76%, a Precision of 99.76%, a Recall of 82.34%, and an F1-Score of 90.21%, indicating a generally balanced performance. In addition, XGBoost achieved the highest discriminative value among all models with a ROC-AUC of 98.35%. This result shows that XGBoost has a strong capacity to distinguish normal and anomalous classes at the probability level.

The SVM algorithm achieved high overall performance with an Accuracy of 97.78% and an F1-Score of 90.25%. However, its ROC-AUC value of 92.64% was lower than those of the stronger models, indicating that the ability of SVM to distinguish classes across different decision thresholds was more limited compared with XGBoost, RF, and DT.

The KNN algorithm produced comparatively lower performance, with an Accuracy of 97.63% and an F1-Score of 89.53%. The distance-based structure of KNN may have limited its ability to learn class distinctions in the CIC-PQC_OAV v1 dataset, which contains high-dimensional and heterogeneous traffic, TLS, and cryptographic context features.

Overall, DT and RF produced the most balanced results in terms of general classification performance and F1-Score. XGBoost, on the other hand, achieved the highest ROC-AUC value and therefore stood out in terms of class separability. Accordingly, RF, DT, and XGBoost can be considered more suitable approaches for anomaly detection on the CIC-PQC_OAV v1 dataset.

The confusion matrices presented in Figure 1 give a better insight into the decision-making process of the models in regard to the normal and anomaly classes. The DT, RF, KNN, and SVM successfully recognized all 7,000 normal instances in the test data and there were no false positives among them. As for the XGBoost model, there were only 2 cases where the normal data was an anomaly. In regard to the anomaly class, the DT, RF, and XGBoost identified 825 instances of anomaly, while SVM recognized 824 and KNN – 812. On the other hand, some anomaly instances were classified as normal traffic by all models. It can be stated that the models are quite effective in avoiding false alarms, but there is one drawback of not identifying all anomaly samples which restricts the Recall values.

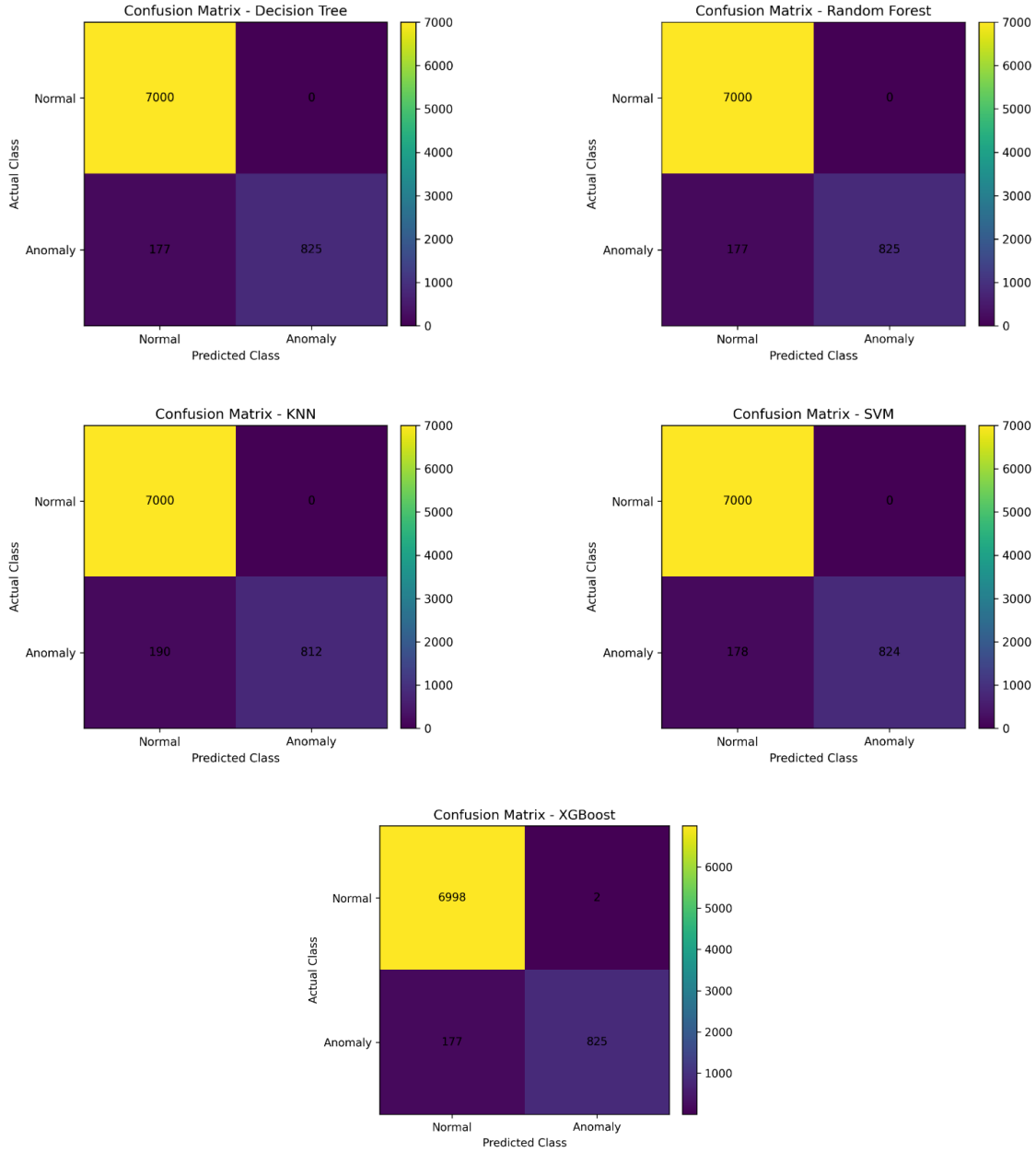


Figure 1. Confusion matrices of the machine learning models

Figure 2 shows the comparison of values for Accuracy, Precision, Recall, F1-Score, and ROC-AUC for the models. From the figure, it is clear that DT and RF offered a fairly balanced score in terms of Accuracy and F1-Score. In terms of ROC-AUC value, the highest was obtained by XGBoost, and therefore it had the best discriminative power among classes. Even though KNN obtained a reasonable score in terms of Accuracy, it was still limited compared to the other models, especially in Recall and F1-Score.

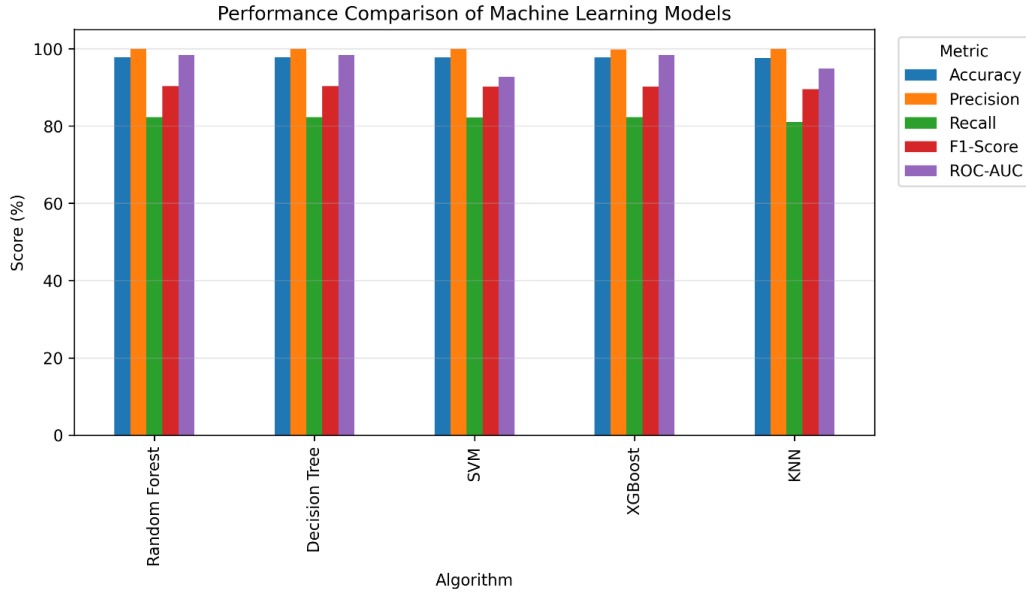


Figure 2. Performance comparison of the machine learning models

Figure 3 shows the ROC curves of the models. The ROC-AUC values indicate the extent to which the models can distinguish between normal and anomalous classes across different decision thresholds. XGBoost reached the highest discriminative performance with an AUC value of 98.35%. RF and DT produced very close results, with AUC values of 98.33% and 98.31%, respectively. Although SVM and KNN produced acceptable results, they remained behind the tree-based and boosting-based methods in terms of ROC-AUC.

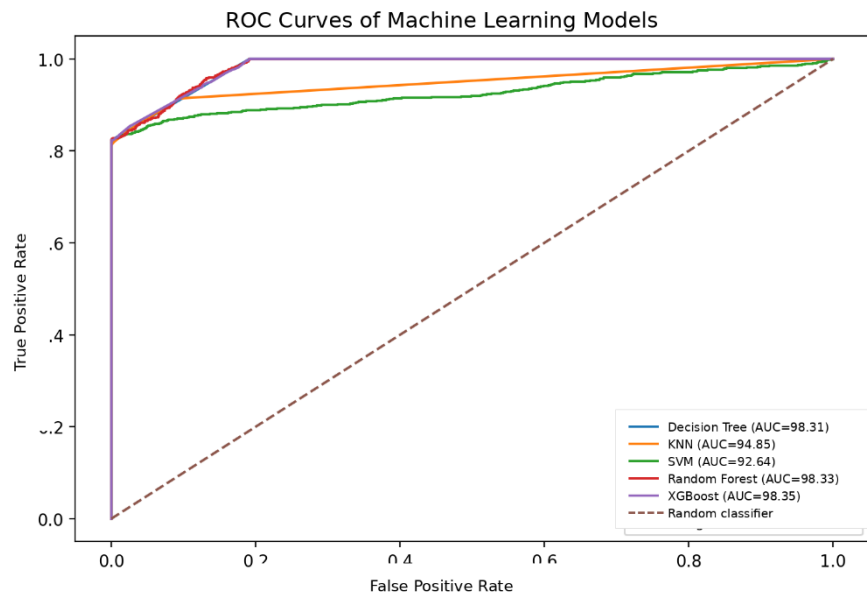


Figure 3. ROC curves of the machine learning models

When the results obtained in this study are evaluated together with studies conducted on traditional IDS datasets in the literature, the accuracy values appear to be slightly below the 98%–99% range reported in some studies. However, this difference should not be interpreted

directly as model failure. The observed differences may be attributed to variations in dataset characteristics, feature composition, traffic patterns, and experimental settings. Therefore, the performance obtained on the CIC-PQC_OAV v1 dataset should be interpreted within the context of its own dataset characteristics rather than through direct comparison with traditional IDS datasets.

In traditional IDS datasets, attack patterns may often be more explicit, whereas in the CIC-PQC_OAV v1 dataset, variables related to traffic volume, TLS record lengths, entropy values, connection outcome, protocol context, and post-quantum key exchange algorithms are evaluated together. Therefore, the reported performance should be interpreted in relation to the characteristics of the CIC-PQC_OAV v1 dataset and the corresponding experimental design. A more appropriate interpretation is to evaluate this value within the specific structure of post-quantum TLS traffic.

In this study, the natural class distribution of the dataset was preserved, and no resampling method such as SMOTE was applied. In addition, fields that could cause data leakage, such as ID, split, and taxonomy, were not included in model training. This choice may have limited the possibility of obtaining higher values in some metrics; however, it allowed the results to be interpreted under more realistic conditions. In this respect, the purpose of the study was not merely to achieve the highest accuracy value but to evaluate the behavior of the models under natural conditions in post-quantum TLS traffic.

The fact that both DT and RF have Precision of 100% suggests that those algorithms do well in avoiding false positives. However, Recall levels that are roughly 82% suggest that there were some anomaly instances labeled as normal. That fact is critical for the safety of the network since false negatives could lead to real anomalies being overlooked. Thus, when assessing machine learning algorithms for post-quantum TLS traffic, one must consider not only Accuracy but also Precision/Recall tradeoff.

The highest ROC-AUC obtained for XGBoost indicates the excellent capability of the classifier to distinguish between the classes based on the probabilities. On the other hand, RF and DT have provided almost comparable outcomes in terms of classification accuracy and F1-Score. It means that the choice of models for the post-quantum TLS traffic should not depend on only one metric.

This study presents a comparative evaluation of machine learning-based anomaly detection in post-quantum TLS 1.3 environments using the CIC-PQC_OAV v1 dataset. Decision Tree, K-Nearest Neighbor, Support Vector Machine, Random Forest, and XGBoost were evaluated using Accuracy, Precision, Recall, F1-Score, and ROC-AUC metrics. The reported results provide an empirical benchmark for anomaly detection in post-quantum TLS traffic and establish a reference point for future comparative studies.

5. CONCLUSION AND FUTURE WORK

This study presents a comparative evaluation of machine learning-based anomaly detection in post-quantum TLS 1.3 environments using the CIC-PQC_OAV v1 dataset. Decision Tree, K-Nearest Neighbor, Support Vector Machine, Random Forest, and XGBoost were evaluated using Accuracy, Precision, Recall, F1-Score, and ROC-AUC metrics. The reported results provide an empirical benchmark for anomaly detection in post-quantum TLS traffic and establish a reference point for future comparative studies.

The results showed that DT and RF models performed best in terms of the measures of classification effectiveness. Both models had an accuracy rate of 97.79% and F1-Score of 90.31%. Also, the fact that the precision rates were 100% in both models is an indication that there were no false positives in the testing set. This could be considered a great advantage in the field of network security systems.

The maximum ROC-AUC value of 98.35% was obtained by the XGBoost algorithm and shows a good capability of distinguishing between normal and abnormal samples based on the probability criterion. Nevertheless, Decision Tree and Random Forest showed the most balanced results in such criteria as Accuracy, Precision, Recall, and F1-Score. In other words, the obtained results imply that the selection of a model for post-quantum TLS traffic analysis should be based on several evaluation criteria but not only on one of them. Although KNN and SVM had quite accurate results as well, they performed less efficiently than the tree-based and boosting-based algorithms in terms of Recall, F1-Score, and ROC-AUC.

In the literature, accuracy values in the range of 98% to 99% are frequently reported on classical datasets such as CICIDS2017, CICIDS2018, CICIoT2023, and UNSW-NB15. In contrast, the highest accuracy value obtained in this study was 97.79%. Therefore, anomaly detection systems developed for post-quantum network environments should not be evaluated solely on the basis of results obtained from traditional IDS datasets.

One of the important contributions of this study is the comparative evaluation of different machine learning algorithms on the CIC-PQC_OAV v1 dataset. In addition, the natural class distribution of the dataset was preserved, resampling methods were not applied, and variables that could cause data leakage were not included in model training. This approach enabled the results to be interpreted in a more realistic and reliable manner.

Several limitations are associated with this study. First of all, only traditional machine learning techniques were tested. The models developed using deep learning approaches, hybrid classification schemes, and explainable artificial intelligence techniques are not included in the scope of this paper. Another limitation lies in the fact that even though the original distribution of classes was maintained, fewer anomalies than normal samples are available, which may have affected Recall. This problem deserves to be analyzed further.

Future research can explore the impact of class balancing techniques, threshold tuning, and feature selection methods on the CIC-PQC_OAV v1 dataset. Additionally, other types of machine learning models like Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), and Autoencoder can also be employed in order to assess the possibility to learn anomaly patterns from post-quantum TLS traffic. Moreover, methods of Explainable Artificial Intelligence (SHAP and LIME) can help in assessing the impact of traffic, TLS, and cryptographic features on model decision making. Thus, the creation of anomaly detection systems which will not only perform well but will also be trustworthy will be facilitated.

References

- [1] K. Sasikumar and S. Nagarajan, “Comprehensive Review and Analysis of Cryptography Techniques in Cloud Computing,” *IEEE Access*, vol. 12, pp. 52325–52351, 2024, doi: 10.1109/ACCESS.2024.3385449.
- [2] E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.3,” Aug. 2018, doi: 10.17487/RFC8446.
- [3] G. M. Raimondo and L. E. Locascio, “Module-Lattice-Based Key-Encapsulation Mechanism Standard,” Aug. 2024, doi: 10.6028/NIST.FIPS.203.
- [4] E. Camacho-Ruiz, P. Navarro-Torrero, and A. C. Aldaya, “A Framework for designing High-Order Side-Channel Protected Hardware Implementations of ML-KEM,” *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2026, no. 2, pp. 272–295, Apr. 2026, doi: 10.46586/TCHES.V2026.I2.272-295.
- [5] B. Yan et al., “Factoring integers with sublinear resources on a superconducting quantum processor,” Dec. 2022, Accessed: Jul. 03, 2026. [Online]. Available: <http://arxiv.org/abs/2212.12372>
- [6] E. Zeydan, Y. Turk, B. Aksoy, and S. B. Ozturk, “Recent Advances in Post-Quantum Cryptography for Networks: A Survey,” *Proceedings of the 2022 7th International Conference on Mobile and Secure Services, MobiSecServ 2022*, 2022, doi: 10.1109/MOBISECSERV50855.2022.9727214.
- [7] D. Joseph et al., “Transitioning organizations to post-quantum cryptography,” *Nature* 2022 605:7909, vol. 605, no. 7909, pp. 237–243, May 2022, doi: 10.1038/s41586-022-04623-2.
- [8] C. Rubio Garcia, A. Cano Aguilera, C. Stan, J. Jose Vegas Olmos, S. Rommel, and I. Tafur Monroy, “Enhanced Network Security Protocols for the Quantum Era: Combining Classical and Post-Quantum Cryptography, and Quantum Key Distribution,” *IEEE Journal on Selected Areas in Communications*, vol. 43, no. 8, pp. 2765–2781, 2025, doi: 10.1109/JSAC.2025.3568011.
- [9] R. Xie et al., “Rosetta: Enabling Robust TLS Encrypted Traffic Classification in Diverse Network Environments with TCP-Aware Traffic Augmentation,” *Proceedings of ACM Turing Award Celebration Conference, CHINA 2023*, vol. 1, pp. 131–132, Jul. 2023, doi: 10.1145/3603165.3607437.
- [10] “PQC-OAV-dataset-2025 | Datasets | Research | Canadian Institute for Cybersecurity | UNB.” Accessed: Jun. 20, 2026. [Online]. Available: <https://www.unb.ca/cic/datasets/pqc-oav-dataset-2025.html>
- [11] H. M. Mujlid and R. Alshahrani, “Quantum-driven security evolution in IoT: AI-powered cryptography and anomaly detection,” *Journal of Supercomputing*, vol. 81, no. 9, pp. 1087–, Jun. 2025, doi: 10.1007/S11227-025-07582-3/TABLES/2.
- [12] K. S. Adewole, A. Jacobsson, and P. Davidsson, “Intrusion Detection Framework for Internet of Things with Rule Induction for Model Explanation,” *Sensors* 2025, Vol. 25, Page 1845, vol. 25, no. 6, p. 1845, Mar. 2025, doi: 10.3390/S25061845.
- [13] M. A. Talukder et al., “Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction,” *J. Big Data*, vol. 11, no. 1, pp. 33–, Dec. 2024, doi: 10.1186/S40537-024-00886-W/TABLES/16.

- [14] S. More et al., “Enhanced Intrusion Detection Systems Performance with UNSW-NB15 Data Analysis,” *Algorithms* 2024, Vol. 17, Page 64, vol. 17, no. 2, p. 64, Feb. 2024, doi: 10.3390/A17020064.
- [15] R. Jablaoui, O. Cheikhrouhou, M. Hamdi, and N. Liouane, “Deep learning enabled intrusion detection system for IoT security,” *EURASIP J. Wirel. Commun. Netw.*, vol. 2025, no. 1, pp. 66-67, Dec. 2025, doi: 10.1186/S13638-025-02477-6/TABLES/8.
- [16] S. Fischer and M. H. Alsulami, “Residual Dense Optimization-Based Multi-Attention Transformer to Detect Network Intrusion against Cyber Attacks,” *Applied Sciences* 2024, Vol. 14, Page 7763, vol. 14, no. 17, p. 7763, Sep. 2024, doi: 10.3390/APP14177763.
- [17] “PQC-OAV-dataset-2025 | Datasets | Research | Canadian Institute for Cybersecurity | UNB.” Accessed: Jul. 03, 2026. [Online]. Available: <https://www.unb.ca/cic/datasets/pqc-oav-dataset-2025.html>
- [18] J. Singh Kushwah, A. Kumar, S. Patel, R. Soni, A. Gawande, and S. Gupta, “Comparative study of regressor and classifier with decision tree using modern tools,” *Mater. Today Proc.*, vol. 56, pp. 3571–3576, Jan. 2022, doi: 10.1016/J.MATPR.2021.11.635.
- [19] R. K. Halder, M. N. Uddin, M. A. Uddin, S. Aryal, and A. Khraisat, “Enhancing K-nearest neighbor algorithm: a comprehensive review and performance analysis of modifications,” *Journal of Big Data* 2024 11:1, vol. 11, no. 1, pp. 113-, Aug. 2024, doi: 10.1186/S40537-024-00973-Y.
- [20] M. Bansal, A. Goyal, and A. Choudhary, “A comparative analysis of K-Nearest Neighbor, Genetic, Support Vector Machine, Decision Tree, and Long Short Term Memory algorithms in machine learning,” *Decision Analytics Journal*, vol. 3, p. 100071, Jun. 2022, doi: 10.1016/J.DAJOUR.2022.100071.
- [21] E. Açıkgozoğlu, “COMPARISON OF MACHINE LEARNING ALGORITHMS FOR DETECTION OF DATA EXFILTRATION OVER DNS,” *Yalvaç Akademi Dergisi*, vol. 9, no. 2, pp. 61–70, Oct. 2024, doi: 10.57120/YALVAC.1507402.
- [22] Z. F. Faruq, T. Mantoro, M. A. Catur Bhakti, and Wandy, “Random Forest Classifier Evaluation in DDoS Detection System for Cyber Defence Preparation,” *2022 IEEE 8th International Conference on Computing, Engineering and Design, ICCED 2022*, 2022, doi: 10.1109/ICCED56140.2022.10010341.
- [23] M. Kumar, C. Kondaiah, A. R. Pais, and R. S. Rao, “Machine learning models for phishing detection from TLS traffic,” *Cluster Comput.*, vol. 26, no. 5, pp. 3263–3277, Oct. 2023, doi: 10.1007/S10586-023-04042-6/TABLES/11.