Sodar et al. *EPJ Quantum Technology* (2026) 13:17
https://doi.org/10.1140/epjqt/s40507-025-00459-7

EPJ.org

EPJ Quantum Technology

RESEARCH Open Access

Encrypted network traffic analysis using quantum machine learning

Gokul Sunil Sodar[1], Akshay Murthy[1], Annapurna Jonnalagadda[1*] and Aswani Kumar Cherukuri[2*]

*Correspondence:
jannapurna@gmail.com;
cherukuri@acm.org
[1]School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, 632014, Tamil Nadu, India
[2]School of Computer Science Engineering and Information Systems, Vellore Institute of Technology, Vellore, 632014, Tamil Nadu, India

Abstract

There is an exponential growth in the encrypted network traffic due to the increased privacy concerns and secure communication needs. This growth has made traditional content- based based traffic analysis techniques ineffective to determine whether traffic is benign or malicious. As a result, security researchers have adopted the practice of examining HTTP header files to analyze encrypted network traffic. This involves inspecting IP addresses, packet sizes, and other metadata. However, these conventional methods face significant limitations in capturing complex temporal dependencies, adapting to evolving threats, and performing, under data sparsity or noise especially in the context of encrypted traffic where visibility is inherently restricted. Traditionally, machine learning and deep learning methods have been successfully used to classify packets as normal or an attack. With the ability to handle high-dimensional and complex nature of this metadata, Quantum Machine Learning (QML) offers a novel paradigm to potentially uncover more intricate patterns that are intractable for classical models. In this paper, we examine the usage of two quantum machine learning models: Quantum Support Vector Machine (QSVM) and Quantum K-Nearest Neighbors (QKNN). We have proposed hybrid quantum machine learning methods, wherein the data is encoded using quantum encodings and then classified using canonical machine learning models. We experimented with different quantum encodings such as angle and amplitude encoding. Our study was conducted on encrypted traffic classification datasets provided by the Canadian Institute of Cybersecurity. Our findings show that the proposed quantum and hybrid quantum models achieve performance comparable to the canonical machine learning models. Notably, the hybrid KNN and SVM models, when paired with amplitude encoding, demonstrated performance on par with or superior to their purely canonical counterparts. We hope that these results would be of interest not only just the researchers and academicians and also practitioners in Cybersecurity industry.

Keywords: Hybrid QML; K-Nearest Neighbors (KNN); Quantum K-Nearest Neighbors (QKNN); Quantum Machine Learning (QML); Quantum Support Vector Machine (QSVM); Support Vector Machine (SVM)

1 Introduction

The usage of encryption protocols has made networks more safe and secure. It has ensured data privacy and security across networks [29]. At the same time, it has also made network

Springer

Table 1 List of Abbreviations

Abbreviation	Full Form
CIC	Canadian Institute for Cybersecurity
CNN	Convolutional Neural Network
CNOT	Controlled-NOT
IDS	Intrusion Detection System
KNN	K-Nearest Neighbors
NISQ	Noisy Intermediate-Scale Quantum
QCNN	Quantum Convolutional Neural Network
QKM	Quantum K-Means
QKNN	Quantum K-Nearest Neighbors
QML	Quantum Machine Learning
QSVM	Quantum Support Vector Machine
RBF	Radial Basis Function
RY	Rotation-Y gate
SVM	Support Vector Machine
URL	Uniform Resource Locator
VQC	Variational Quantum Classifier

traffic analysis more challenging as analysis of packets has to be solely done on metadata such as packet size, timing, and other flow characteristics [2, 9, 12]. Many machine learning and deep learning approaches have been used to perform analysis of encrypted network traffic. They use statistical or behavioral features like flow duration, packet size and other such features to analyze the packets. [29]. Although such methods yield good results, they might suffer from the "Curse of Dimensionality" when analysing encrypted network traffic [2]. Moreover, these classical approaches face challenges in adapting to adversarial behaviors, coping with sparse or noisy metadata, and capturing long-range temporal dependencies in encrypted flows. This motivates the exploration of alternative computational paradigms, particularly those capable of handling high-dimensional representations more efficiently.

Quantum computing is a computational paradigm that uses the principles of quantum mechanics to solve problems that are intractable for classical computers. While classical computers are limited by bits representing either 0 or 1, the power of quantum computing stems from its use of qubits. Due to superposition, a qubit can represent a combination of 0 and 1 simultaneously, allowing a quantum computer with n qubits to explore a computational state space of 2^n dimensions. This exponential scaling in information capacity gives quantum computers a significant advantage for tackling certain complex problems. In the context of machine learning, this ability to map classical data into exponentially large Hilbert spaces enables richer feature representations than what is feasible with conventional kernels or embeddings. Thus, quantum computing offers a new computational tool for analyzing encrypted network traffic.

This inherent power is what makes QML a compelling evolution of classical machine learning [23]. The primary advantage QML can provide over traditional ML is the ability to use this vast quantum state space as a feature space. By encoding classical information into quantum states using qubits [26], QML models can analyze data in exceptionally high dimensions. This allows them to use the principles of superposition and entanglement to explore complex patterns and correlations within the data far more efficiently than classical methods, offering a new way of analyzing data that would otherwise be computationally prohibitive [5, 17].

Encrypted network traffic analysis relies on statistical and temporal features extracted from metadata, which can be high-dimensional in nature. QML's nature to work well in high-dimensional spaces makes it an interesting tool to use when analysing encrypted network traffic. Quantum machine learning models have shown that they can improve performance compared to classical machine learning models. [11, 13, 20, 24, 32].

QSVM and QKNN are QML classifiers worth investigating for encrypted traffic classification as they have the potential to deliver improved performance when classifying complex data [16]. QSVM is a compelling candidate because it leverages the quantum kernel trick to map data into an exponentially large feature space, potentially identifying complex, non-linear decision boundaries that are inaccessible to classical kernels. QKNN is also promising as it uses a quantum distance metric computed directly in the Hilbert space, which may provide a more meaningful measure of similarity for distinguishing between nuanced classes of encrypted traffic compared to classical distance metrics.

QSVM and QKNN use quantum circuits and distance metrics to construct their respective decision boundaries. Due to the nature of quantum circuit simulation and quantum distance computation, QSVM and QKNN take a longer time to train and test data when run on classical simulators. To address this limitation, we propose using a hybrid QML approach where we combine quantum feature encoding with regular machine learning classifiers. The hybrid QML models retain the expressive nature of quantum circuits while also benefiting from the speed of the regular machine learning classifiers.

The primary contribution of this work is the design and evaluation of a hybrid QML framework for the classification of encrypted network traffic. We define our "Hybrid QML" approach as a two-stage process: first, classical network data is transformed and mapped into a quantum feature space using quantum encoding techniques; second, canonical machine learning algorithms—specifically SVM and KNN are trained on these quantum-encoded features to classify packets as benign or malicious.

Following this core contribution, we conduct a comprehensive comparative analysis. This study systematically compares the performance of our hybrid QSVM and hybrid QKNN models. A key part of this comparison is assessing the impact of different quantum feature representations, providing insights into the relative effectiveness of angle encoding versus amplitude encoding for this specific cybersecurity task. Through this analysis, our work offers valuable insights into the practical application and effectiveness of hybrid QML models in network security.

Unlike existing works, our contribution lies in contextualizing QML performance specifically for encrypted network traffic classification. We highlight both the feasibility and the constraints of such approaches, including the reliance on simulators, scalability limitations with high-dimensional encodings, and the modest nature of observed performance gains. We emphasize that our results demonstrate the practicality of hybrid QML as a complementary tool that can inform future quantum ready Cybersecurity solutions.

While prior works such as Guerrero-Estrada et al. (2024) have benchmarked QKNN with amplitude encoding for network tasks, and Medeiros de Abreu et al. (2025) have conducted comparative evaluations of several QML approaches across multiple datasets, these studies do not focus specifically on encrypted network traffic classification.

Moreover, they do not systematically examine how different encoding strategies (e.g., angle versus amplitude) interact with both quantum and classical classifiers in a hybrid setting. Our work addresses this gap by conducting a focused, application-driven evalu-

ation of hybrid QML models on encrypted traffic datasets, thereby providing empirical insights tailored to cybersecurity contexts.

Contributions of this study:

- We proposed a hybrid quantum - classical ML model for analysing the encrypted network traffic.
- The proposed model leverages canonical machine learning algorithms on top of quantum feature encoding techniques for classification of encrypted network traffic packets as benign or malicious.
- Performance of the proposed hybrid QML method that applies canonical machine learning models to quantum encoded encrypted network traffic data was examined using both angle and amplitude encodings on the ISCX-URL2016 and ISCX-IDS2012 datasets.
- Further, we have also evaluated the performance of QSVM and QKNN models with both angle and amplitude encodings on the same datasets.

This work should be seen as a focused, ENTA oriented complement to existing QML-for network-security studies such as QuantumNetSec (Medeiros de Abreu et al., 2025)., rather than as a competing framework. Our contribution lies in a systematic, encoding-centric comparison of classical, pure quantum, and hybrid SVM/KNN–QML models for encrypted network traffic on CIC-style datasets, with particular attention to angle versus amplitude encodings, qubit requirements, and practical simulation overhead.

Rest of the paper is organized as follows. Section 2 provides background and related work. Section 3 provides the methodology we have adopted. Section 4 provides the details of the experimental analysis followed by the discussion, conclusions and references.

2 Background and related work

2.1 Encrypted network traffic classification

Encryption has enhanced security in networks, but at the same time, it has also made network traffic classification difficult. Methods like deep packet inspection have become ineffective as they cannot access payload data [30]. Encrypted network traffic classification now relies on extracting statistical features such as packet sizes, flow durations, inter-arrival times and other such features. Other features like port numbers, IP addresses, and TCP flags are also used [2]. Many machine learning and deep learning models have been used for analysing encrypted network traffic [22, 29]. These models analyze the packets to capture temporal and behavioral patterns. This helps to differentiate between the different types of encrypted traffic.

Traditional machine learning classifiers have been used for encrypted traffic classification. [30] provides a structured review of machine learning methods used to detect malicious activity hidden within encrypted network traffic. They evaluated ten machine learning algorithms across their dataset. Their comparative study identifies the best performing algorithms and feature sets. Their study showed that while deep learning is gaining traction, traditional ML still offers strong results.

Recent approaches have also focused on using deep learning architectures to model encrypted traffic more effectively. These models automatically learn hierarchical representations from raw or minimally processed traffic features. [22] introduced Deep Packet, a framework using CNNs to classify encrypted traffic with high accuracy. [27] demonstrated the strength of deep autoencoders in identifying traffic patterns without relying on payload data. [15] proposed TransNet, a framework that enables collaborative neural network

training for classification tasks without revealing participants' raw data. Instead of relying on traditional cryptographic methods like homomorphic encryption, TransNet employs a transformed layer that irreversibly obfuscates each participant's data before sharing it for model training. This approach preserves the utility of the data for neural network classification while ensuring privacy. This also supports various data partitioning scenarios, incurring minimal computational and communication overhead compared to existing privacy preserving methods.

Existing research has also explored the application of federated learning and privacy preserving techniques to traffic classification. This aims to balance model performance with user privacy. [18] introduces FedETC, a federated learning framework designed for encrypted network traffic classification. Rather than relying on manual feature engineering, the authors deploy a 1D convolutional neural network that learns directly from byte level encrypted traffic. FedETC enables multiple network domains to train a global model using the FedAvg algorithm. This ensures local data privacy without sharing raw encrypted traffic. Experimental evaluation on real world encrypted datasets shows that FedETC achieves accuracy comparable to centralized approaches.

2.2 Quantum computing

Quantum computing represents a fundamental shift from classical computation, leveraging the principles of quantum mechanics to process information in profoundly new ways. Unlike a classical bit, which is restricted to a definite state of either 0 or 1, a qubit can exist in a superposition of both states simultaneously [8]. For QML, this property is transformative: it allows for the creation of exponentially large feature spaces. A system of just n qubits can represent 2^n classical values at once, providing a vast arena in which to encode complex datasets and search for subtle patterns [4, 26].

This potential is further amplified by entanglement, a non-classical correlation where the state of one qubit is intrinsically linked to another, regardless of the physical distance separating them [8, 26, 31]. In the context of QML, entanglement is a powerful resource for modeling complex relationships and correlations between data features. By entangling qubits that represent different features, a QML model can capture intricate, high-order dependencies that may be computationally prohibitive for classical models to learn.

These quantum states are manipulated using quantum gates, which perform precise unitary operations on the qubits [10]. In a QML algorithm, these gates are arranged into a quantum circuit, which acts as the core processing engine. The circuit is strategically designed to perform tasks such as mapping classical data into the quantum feature space (feature mapping) or calculating the similarity between two data points (the kernel trick). Collectively, the ability of superposition to create vast computational spaces, the power of entanglement to model complex correlations, and the precise control offered by quantum circuits are what give QML algorithms their potential to deliver significant computational advantages over their classical counterparts [3, 26].

2.3 Quantum machine learning

As classical machine learning models encounter fundamental limits in computational power and struggle with increasingly complex, high-dimensional datasets, QML presents a necessary and promising alternative [28]. The core advantage of QML over its classical counterpart is its ability to harness quantum phenomena. By utilizing qubits, which

can exist in a superposition of states, QML algorithms can process information in exponentially large computational spaces known as Hilbert spaces [26]. Mapping classical data into these spaces enables the design of quantum-enhanced machine learning classifiers that can, in principle, identify patterns that are invisible to classical algorithms [19].

However, the current generation of quantum hardware, known as NISQ technology, is not yet capable of running fully quantum algorithms flawlessly. This is where a hybrid QML approach becomes essential. Hybrid models strategically divide labor: the bulk of the workflow, such as data preparation and model optimization, is run on robust classical computers, while a specific, computationally-intensive task is offloaded to a quantum processor. This allows researchers to leverage the unique capabilities of quantum computation for critical steps (like feature mapping or kernel evaluation) while mitigating the impact of hardware noise and limitations, thereby providing a practical pathway to explore quantum advantages today.

[26] applied various quantum encodings on data in a customer churn dataset. They primarily investigated basis, angle and amplitude encodings. Their findings demonstrate that quantum data embeddings can improve classification accuracy and F-1 scores for models that benefit from enhanced feature representation. The ensemble classifiers they used showed a balance between performance gains and computational overhead.

The selection of appropriate encoding methods is very important for performing classification. [13] found that amplitude encoding showed promising results when integrated with a QKNN implementation achieving a 50% reduction in required qubits while maintaining similar overall performances.

2.4 QSVM and QKNN

Quantum support vector machines can use quantum kernel methods to give better performance on high-dimensional data than classical SVMs [32]. By mapping classical data into quantum Hilbert spaces using quantum circuits, QSVM can achieve noise resilience [11]. Further research has focused on refining these models; for example, [25] discussed how to create more robust decision boundaries and prevent overfitting by proposing the use of unitary transformations with rotation factors and added regularization. Yin [32] explored QSVM fundamentals. Her work talks about quantum state preparation and performing kernel computation using quantum circuits. The study highlights the theoretical advantages of using QSVM over classical SVM. When applied to network traffic analysis, Kalinin and Krundyshev [20] demonstrated that QSVM can outperform classical SVM for detecting network attacks.

[13] implemented a QKNN classifier and applied it across 13 datasets. They were able to achieve an increase in accuracy and F1-scores for some of their datasets. [11] implemented a QKNN classifier for forensic document analysis. They utilized a grey level co-occurrence matrix for feature extraction, which is robust to changes in rotation and scale. They demonstrated that QKNN outperforms classical KNN in terms of accuracy when identifying subtle printing artifacts. [33] implemented a QKNN algorithm using Euclidean distance. Bhaskaran and Prasanna [5] demonstrated the potential to achieve high accuracies using QKNN when trying to predict breast cancer. They were able to achieve a high accuracy of 93.85%.

[14] proposed three novel quantum autoencoder based anomaly detection frameworks.. They used quantum autoencoders with quantum random forest, quantum k-nearest

neighbors and quantum one-class support vector machines. They applied this on a network traffic dataset and demonstrated that all 3 frameworks can effectively detect network traffic anomalies with the QKNN framework giving the highest accuracy.

Abreu et al. introduced QuantumNetSec, a QML-based IDS designed for current NISQ devices. [24]. Their work provides a comprehensive analysis by testing four distinct QML techniques namely VQC, QSVM, QCNN, and QKM across four major network security datasets, including UNSW-NB15 and TONIoT. They implemented and evaluated personalized circuit optimization strategies, which showed improvement in model performance on real quantum hardware. Their findings show that QML models, particularly QCNN, can outperform classical counterparts in both binary and multiclass attack classification.

The use of QML to potentially provide better performance than regular machine learning algorithms inspired us to investigate its usage for encrypted network traffic analysis.

- QML has been explored in the field of Encrypted Network Traffic Analysis. However, the research is still in its early stages. We believe that further investigation is needed to understand its full potential.
- Hybrid quantum machine learning techniques have not been explored enough in the field of Encrypted network traffic analysis. Our work focuses on their applicability and advantages of the hybrid models in this domain.

3 Methodology

We propose three methodologies to analyse encrypted network traffic: kernel-based QSVM, QKNN and hybrid QML.

3.1 QSVM

For QSVM, we used the classical SVM function, but instead of traditional kernels such as linear or radial basis function kernels, we employed a quantum kernel function [32]. Similar to how non-linear kernels like polynomial and RBF use the kernel trick, a similar quantum kernel trick was employed here by utilising a quantum circuit inside the quantum kernel function. In essence, the classical kernel trick allows an algorithm to learn in a high-dimensional feature space by only computing the inner product between data points, avoiding the computationally expensive step of explicitly mapping the data into that space. Analogously, our quantum kernel uses a quantum circuit to efficiently calculate the inner product (or overlap) between data points in an exponentially large quantum feature space, a task that would be intractable for a classical computer. The architecture of QSVM is illustrated in Fig-1.

The quantum circuit implements this by first converting two samples, x_i and x_j, into quantum embeddings, $|\phi(x_i)\rangle$ and $|\phi(x_j)\rangle$. For the embeddings, we utilised angle and amplitude encodings. The circuit then computes the overlap between these states, with the kernel value being the squared magnitude of this inner product:

$$K(x_i, x_j) = |\langle\phi(x_i)|\phi(x_j)\rangle|^2 \tag{1}$$

These kernel values for all data pairs are used to construct the final gram matrix. To optimize this computation, we focused on calculating only the lower triangular portion of the gram matrix. The remaining values were then copied to complete the matrix, reducing

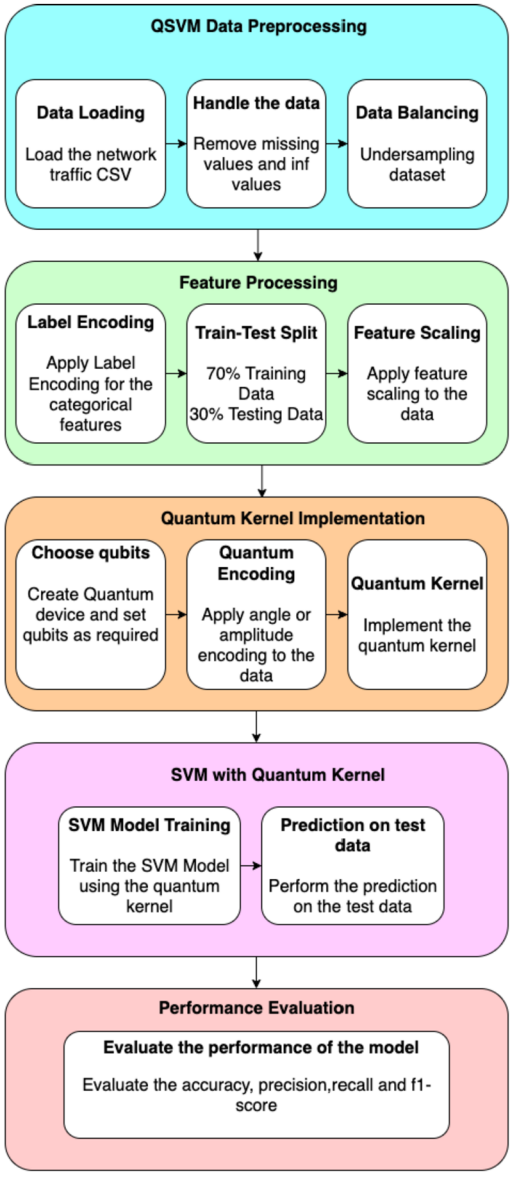


Figure 1 QSVM Architecture Diagram

computations and speeding up the process. Additionally, to further accelerate the computation, we parallelised the kernel matrix calculation. Algorithm 1 presents the proposed QSVM for encrypted network traffic analysis.

3.2 QKNN

In QKNN, each training and test sample is transformed into a quantum state using either an angle or amplitude embedding function. After this transformation, the inner product is computed between the quantum state of each test sample and the quantum states of all training samples. The square of the inner product is taken to compute the fidelity between the quantum states, which ranges from 0 to 1. The distance is then computed using the

Algorithm 1 QSVM for Encrypted Network Traffic Classification

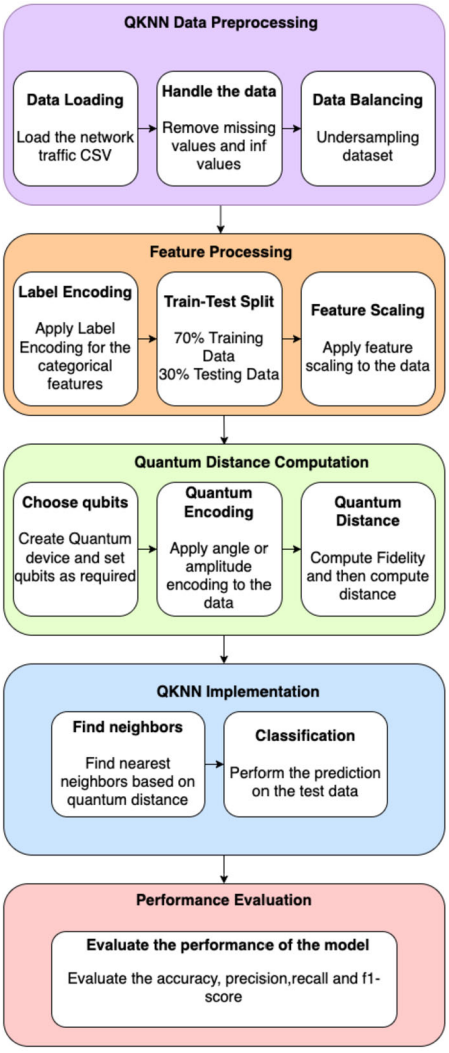
Require: Preprocessed training data X_{train}, training labels y_{train}, test data X_{test}, number of qubits n
Ensure: Predicted labels y_{pred} for test data
function QuantumKernel(a, b)
    Initialize quantum device with n qubits
    Apply quantum encoding of vector a into quantum state
    Apply adjoint of quantum encoding of vector b
    Measure probability of all qubits being in state $|0\rangle$
    return Measured probability $p(|0\rangle)$
end function
function OptimizedQuantumKernelMatrix(A, B)
    Initialize Gram matrix K of size $|A| \times |B|$
    if A and B are identical **then**
        Compute lower triangle of K using parallel processing:
        for each pair (i, j) where $i \geq j$ **do**
            $K_{i,j} \leftarrow$ QuantumKernel(A_i, A_j)
            if $i \neq j$ **then**
                $K_{j,i} \leftarrow K_{i,j}$ ▷ Use symmetry for upper triangle
            end if
        end for
    else
        Compute full matrix K using parallel processing:
        for each pair (i, j) **do**
            $K_{i,j} \leftarrow$ QuantumKernel(A_i, B_j)
        end for
    end if
    return K
end function
function QSVM($X_{train}, y_{train}, X_{test}$)
    Initialize SVM classifier with OptimizedQuantumKernelMatrix as kernel
    Train SVM: $model \leftarrow$ SVM.fit(X_{train}, y_{train})
    Generate predictions: $y_{pred} \leftarrow model$.predict($X_{test}$)
    return y_{pred}
end function
$y_{pred} \leftarrow$ QSVM($X_{train}, y_{train}, X_{test}$)
Evaluate (accuracy, precision, recall, F1-score)

following formula:

$$\text{distance} = 1 - \text{fidelity} \tag{2}$$

This becomes the quantum version of the distance metric used in classical k-Nearest Neighbors. Then the distances are sorted and the top k nearest neighbours are chosen. Fi-

Figure 2 QKNN Architecture Diagram



nally, we take the majority label among the nearest neighbours and return it. The motivation to use fidelity for our QKNN classifier comes from [1]. Fig2 illustrates the architecture of the QKNN.

Algorithm 2 discusses the proposed QKNN methodology for encrypted network traffic analysis.

3.3 Hybrid QML

Hybrid QML is an emerging field that combines classical machine learning algorithms with the principles of quantum computing. This approach leverages quantum mechanics to potentially accelerate or improve upon classical machine learning techniques. In our proposal, the novelty is on realizing the hybridization through quantum encodings of the encrypted network traffic data and implementing canonical and quantum ML models on these encodings. The methodology [26] starts with the preprocessed classical data being mapped into a quantum state using quantum feature encoding techniques. Specifically, we explore two distinct methods for this mapping: angle encoding and amplitude encoding. Once the data is represented in the quantum feature space, the classification is performed

Algorithm 2 QKNN for Network Traffic Classification

Require: Network traffic dataset with features X and labels y
Require: Number of qubits n for quantum processing
Require: Number of neighbors k for KNN algorithm
function QuantumKNN($n\_neighbors, n\_qubits$)
  Initialize quantum device with n_qubits qubits
  Define state preparation circuit to encode data points into quantum states
  function QuantumDistance(x_1, x_2)
    Apply quantum encoding to x_1 and obtain quantum state $|\psi(x_1)\rangle$
    Apply quantum encoding to x_2 and obtain quantum state $|\psi(x_2)\rangle$
    Calculate fidelity between states: $F = |\langle\psi(x_1)|\psi(x_2)\rangle|^2$
    Compute quantum distance as $d = 1 - F$
    return d
  end function
  function Fit(X_train, y_train)
    Store training data X_train and labels y_train
    return *self*
  end function
  function Predict(X_test)
    Initialize empty array y_pred
    for each test sample $x\_test \in X\_test$ **do**
      Initialize empty array *distances*
      for each training sample $x\_train \in X\_train$ **do**
        d = QuantumDistance(x_test, x_train)
        Append d to *distances*
      end for
      Find indices of k smallest distances: *indices* = argsort(*distances*)[: k]
      Extract classes of k nearest neighbors: *neighbors* = $\{y\_train[i] \mid i \in indices\}$
      Determine majority class via voting: *prediction* = mode(*neighbors*)
      Append *prediction* to y_pred
    end for
    return y_pred
  end function
end function
function Main($filepath, n\_qubits, k$)
  Load and preprocess network traffic data
  Split data into training and testing sets
  Scale features and normalize to the appropriate range
  Apply quantum encoding to transform classical data to quantum representation
  Initialize QKNN model with k neighbors and n_qubits qubits
  Sample subset of training data to manage computational complexity
  Train model: qknn.fit($X\_train\_quantum, y\_train$)
  Make predictions: y_pred = qknn.predict($X\_test\_quantum$)
  Evaluate model performance (accuracy, precision, recall, F1, etc.)
  return Performance metrics
end function

using well-established classical algorithms, namely SVM and the KNN classifier. This hybrid strategy allows us to assess the impact of quantum feature spaces on classical algorithms and to perform a comparative analysis between the two encoding methods. The working principles of these two encoding strategies are detailed in the following subsections. The hybrid model architecture diagram is depicted in Fig3.

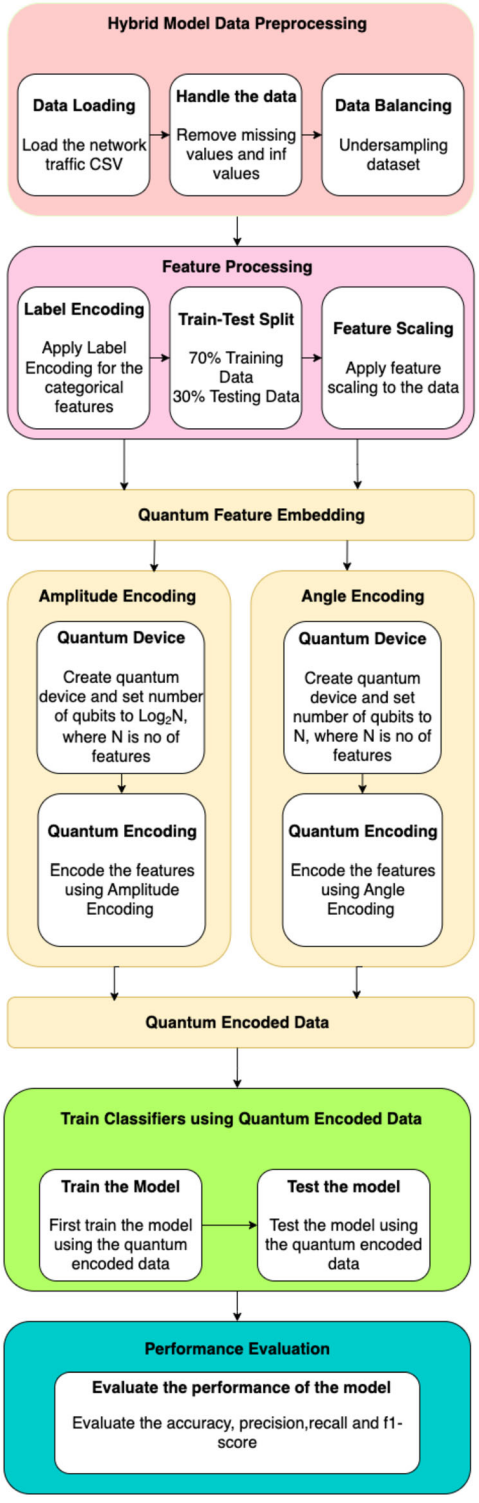


Figure 3 Hybrid Model Architecture Diagram

3.3.1 Amplitude encoding

Amplitude encoding involves normalizing the data row-wise so that the sum of the squares of the amplitudes adds up to one. The number of qubits needed for encoding a sample with n features would be $\log_2(n)$

$$\boldsymbol{\alpha} = (\alpha_0, \ldots, \alpha_{2^N-1}) \mapsto \sum_{k=0}^{2^N-1} \alpha_k \, |k\rangle \tag{3}$$

3.3.2 Angle encoding

Angle encoding involves encoding a single floating-point value into a quantum state by applying a rotation gate.

$$R_\phi(x) = e^{-i\frac{x}{2}\sigma_\phi} \tag{4}$$

where $x \in \mathbb{R}$ is the floating-point value to be encoded, and σ_ϕ is the axis of rotation in the Bloch sphere, which can be the Pauli matrix σ_x, σ_y, or σ_z corresponding to rotation around the x-, y-, or z-axis respectively.

In our implementation, angle encoding uses one qubit per feature, so an input vector with d features requires d qubits. In contrast, amplitude encoding maps an entire d-dimensional feature vector into the amplitudes of a single quantum state, requiring $log_2 d$ qubits. For example, 16 features need 16 qubits with angle encoding but only 4 qubits with amplitude encoding, while 1024 features would still require only 10 qubits with amplitude encoding.

The qubit efficiency of amplitude encoding comes at the price of more complex state preparation. Preparing a general d-dimensional amplitude-encoded state typically requires circuits whose size (and number of entangling gates) scales on the order of $O(d)$, or equivalently $O(2^n)$ in the number of qubits $n = \lceil log_2 d \rceil$, in the worst case.

As a result, while amplitude encoding is very attractive for moderate feature dimensions (tens to a few hundreds of features), for very high-dimensional inputs (hundreds to thousands of features) the qubit savings can be offset by the growth in circuit depth and gate count, especially on NISQ hardware where noise limits usable depth. In such regimes, practical deployments often rely on some combination of classical dimensionality reduction, sparse/state-structured encoders, to keep the overhead manageable.

Beyond the angle and amplitude encodings studied here, several other encoding strategies appear promising for encrypted network traffic classification. Basis (computational) encoding offers a natural way to represent binary header and flag information, but it scales linearly in qubit count and by itself does not exploit richer feature maps. Hamiltonian and IQPstyle encodings, in which data parameters control diagonal unitaries or instantaneous quantum polynomial (IQP) circuits, can generate highly non-linear feature spaces and have been shown to induce distinct topological structures compared to simple angle embeddings. In this work, we focused on angle and amplitude encodings as widely used, hardware-agnostic baselines for a first ENTA study. The hybrid QML algorithm is presented in Algorithm 3.

Algorithm 3 Hybrid QML Algorithm (Quantum Embedding)

Require: Raw dataset with features and labels
Ensure: Trained hybrid quantum-classical classifiers with performance metrics
Data Preprocessing:
Load dataset from CSV file
Remove duplicate records and unwanted columns
Apply label encoding to categorical features
Balance dataset using undersampling techniques
Split data into training and testing sets: (X_{train}, y_{train}), (X_{test}, y_{test})
Transform labels: $y \leftarrow 2(y - 0.5)$ ▷ Convert to $\{-1, +1\}$
Apply scaler normalization to features
Quantum Circuit Definition:
function QuantumEmbedding($\mathbf{x}$)
    Initialize quantum device with n qubits where $n = |\mathbf{x}|$
    Apply quantum embedding of classical data $\mathbf{x}$
    return Expectation values $\langle Z_i \rangle$ for $i \in \{1, \ldots, n\}$
end function
Quantum Feature Transformation:
function QuantumTransform(X)
    Initialize empty array $X_{quantum}$
    for each sample $\mathbf{x}$ in X **do**
        $\mathbf{f} \leftarrow$ QuantumEmbedding($\mathbf{x}$)
        Append $\mathbf{f}$ to $X_{quantum}$
    end for
    Convert $X_{quantum}$ to DataFrame with quantum feature names
    return $X_{quantum}$
end function
Hybrid Model Training and Evaluation:
$X_{train}^{quantum} \leftarrow$ QuantumTransform(X_{train})
$X_{test}^{quantum} \leftarrow$ QuantumTransform(X_{test})
SVM Classification:
Initialize SVM classifier $\mathcal{M}_{SVM}$
Train $\mathcal{M}_{SVM}$ using $(X_{train}^{quantum}, y_{train})$
$\hat{y}_{SVM} \leftarrow \mathcal{M}_{SVM}$.predict($X_{test}^{quantum}$)
Compute SVM accuracy and classification report
KNN Classification:
Initialize KNN classifier $\mathcal{M}_{KNN}$ with $k = 7$, Euclidean distance, distance weighting
Train $\mathcal{M}_{KNN}$ using $(X_{train}^{quantum}, y_{train})$
$\hat{y}_{KNN} \leftarrow \mathcal{M}_{KNN}$.predict($X_{test}^{quantum}$)
Compute KNN accuracy and classification report
return Trained models $\{\mathcal{M}_{SVM}, \mathcal{M}_{KNN}\}$ and performance metrics

4 Experimental setup

4.1 Dataset acquisition

The datasets we used were obtained from the Canadian Institute for Cybersecurity (CIC) situated at the University of New Brunswick. We used the URL dataset (ISCX-URL2016) and the Intrusion Detection Evaluation dataset (ISCXIDS2012) for our study. These datasets consist of both normal and benign traffic, proving useful for testing our models.

The URL dataset (ISCX-URL2016) [7] was developed by the Canadian Institute for Cybersecurity to research URL classification. The dataset contains 15 features. The collection of URLs were either labeled as benign or malicious. We dropped all duplicate records and any rows containing missing values. There was a slight imbalance present in the dataset after the preprocessing stage. To handle this, we undersampled the benign class to match the number of records present in the defacement class. No categorical features were present in the dataset. Feature scaling was applied to the dataset to make it suitable for the machine learning algorithms we used.

The intrusion detection evaluation data set (ISCXIDS2012) [6] was also developed by the Canadian Institute of Cybersecurity. The dataset was generated in a controlled testbed environment. The dataset consists of 20 features. During the preprocessing phase, many columns were discarded due to their unsuitability for machine learning. These columns contained a high proportion of missing values and information that was not structured or interpretable for machine learning algorithms. The dataset contains two primary labels: normal and attack. Duplicate records and rows with missing values were removed, The normal class was undersampled to match the number of records of the attack class. There were a few columns containing categorical values. The categorical features present in the dataset were converted into numerical form using label encoding. We also used feature scaling in this dataset to make it suitable for our machine learning models.

4.2 Learning and preparation phase

4.2.1 Classical ML baselines

The classical machine learning algorithms that we used are SVM and KNN. We utilised scikit-learn's SVC and KNeighborsClassifier functions to run SVM and KNN on both datasets. The preprocessing remained largely the same except when it came to scaling, where we used StandardScaler for the classical SVM and KNN functions.

In this study, we focused our classical baselines on SVM and KNN rather than including deep learning models. While there is a growing body of work on QML and hybrid quantum–classical approaches for network intrusion detection, these studies mainly target generic or IoT traffic. But to the best of our knowledge, they do not provide a systematic analysis of classical ML models hybridised with quantum encodings specifically for encrypted network traffic on CIC-style datasets. Our choice of SVM and KNN, in combination with quantum encodings, is therefore motivated by the aim to help fill this gap in the ENTA literature.

4.2.2 Hybrid QML approach

This implementation was done using Python. The pre-processed data is converted into quantum encodings using quantum circuit functions. These quantum circuit functions were implemented using Pennylane. Pennylane is a Python library used to simulate quantum circuits. We used the lightning.qubit device in Pennylane for faster fit and predict

times. The lightning.qubit device is based on a C++ backed simulator that is faster than the default python-based simulators. We utilised both angle and amplitude encodings. The quantum encoded data was then fed into the SVM and KNN model for the training and testing. We implemented SVM and KNN using scikit-learn's SVC and KNeighborsClassifier functions, respectively.

4.2.3 QSVM

The QSVM implementation was carried out using scikit-learn's SVC function however, instead of using any default kernel for the SVC function, we used a custom quantum kernel. The quantum kernel computes the overlap between two inputs. The quantum kernel function consists of a quantum circuit made using Pennylane. The quantum kernel function accepts two arguments; the quantum encoding function is applied on one input and the adjoint (inverse) of the encoding function is applied on the other input. This overlap is evaluated using the probability of measuring the all-zero computational basis state. To accelerate the quantum kernel function, we parallelized the computation. The quantum kernel function can't be parallelized in a GPU because the lightning.gpu device wasn't picklable. So instead, we used the lightning.qubit device that runs on CPU with a C++ backend and parallelly computed the kernel values by using all CPU cores using python's joblib library. We utilized the Parallel and delayed functions from the Joblib library to parallelize this computation, which significantly reduced fit and predict time. Since the Gram matrix is a square matrix, we only computed the lower part of the matrix and replicated the values for the upper part to further reduce the amount of computation done for the fit function. The default kernels had built-in optimization, but since we implemented a custom kernel, we had to manually configure optimization, which wasn't included by default.

4.2.4 QKNN

In QKNN implementation, after preprocessing max absolute scaler was used. The scikit-learn's MaxAbsScaler converts the data into a range of -1 to 1. The data is clipped to make it in the range of 0 to 1. This was done as it is good practice since amplitude encoding requires the data to be in the range of 0 to 1. QKNN functions were implemented from scratch due to the lack of built-in libraries for QML algorithms. The QKNN fit function converts the entire dataset into quantum embeddings. We utilised both angle and amplitude embeddings. Another custom quantum circuit was also used when the fidelity was computed. This circuit first applies encoding function, followed by two layers of entanglement using CNOT gates and fixed RY gate, before measuring the state. This circuit enables richer quantum representations and is used within the fidelity function to compute state overlaps between test and training samples. To accelerate the fidelity function, we utilised the Parallel and delayed functions from the Joblib library to parallelize this computation which reduced predict time. The fidelity value is used to determine the quantum distance, which is then used for classification via majority voting. The fidelity function was implemented using the NumPy library, and Pennylane was used for quantum circuit implementation. We used the lightning.qubit device in Pennylane for faster execution.

5 Evaluation and comparative analysis

5.1 Evaluation metrics

As indicated above, the performance of the proposed models was evaluated on two encrypted network traffic datasets: ISCX-URL2016 and ISCXIDS2012. Both these datasets

contain flows labeled as benign or malicious, and are widely used for benchmarking in the encrypted traffic classification domain. Although these datasets are widely referenced, we acknowledge that they are relatively limited in diversity and size compared to more recent benchmarks such as CICIDS2017 or UNSW-NB15. As such, our evaluation should be regarded as a feasibility demonstration rather than a conclusive benchmark.

We use the following evaluation metrics to evaluate our models: Accuracy, Precision, Recall and F1-Score.

1. Accuracy

Accuracy is the proportion of all classifications that were correct, whether positive or negative.

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

Where:

TP=True Positives

TN=True Negatives

FP=False positives

FN=False Negatives

2. Precision

Precision is the proportion of true positive predictions out of all positive predictions made by the model.

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

3. Recall

The true positive rate (TPR), or the proportion of all actual positives that were classified correctly as positives, is also known as recall.

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

4. F1-Score

F1-Score is the harmonic mean of precision and recall. A higher F1-Score indicates better performance.

$$\text{F1-Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

5.2 Comparative results

Tables 2 and 3 present the performance of classical baselines (SVM, KNN), quantum variants (QSVM, QKNN), and hybrid QML models with angle and amplitude encoding. Several conclusions can be drawn from the results. The hybrid KNN and SVM with amplitude encoding performed broadly comparable to their classical counterparts. While marginal improvements (e.g., $\leq 0.5\%$ in ISCX-URL2016) were observed in some cases, these differences are modest and dataset-specific rather than universally superior Hybrid SVM gave an accuracy of 0.9667 and hybrid KNN gave an accuracy of 0.9878 as opposed to 0.9634 and 0.9874 of regular SVM and KNN respectively in the ISCX-URL2016 Dataset.

Table 2 Performance Comparison on ISCX-URL2016 Dataset

Model Type	Variant	Accuracy	Precision	Recall	F1-Score
SVM	Regular	0.9634	0.9857	0.9424	0.9635
	Quantum (Angle)	0.9201	0.9207	0.9208	0.9201
	Quantum (Amplitude)	0.9179	0.9178	0.9180	0.9179
	Hybrid (Angle)	0.9423	0.9674	0.9186	0.9424
	Hybrid (Amplitude)	0.9667	0.9880	0.9467	0.9669
KNN	Regular	0.9874	0.9878	0.9878	0.9878
	Quantum (Angle)	0.9645	0.9600	0.9673	0.9637
	Quantum (Amplitude)	0.9471	0.9399	0.9521	0.9460
	Hybrid (Angle)	0.9808	0.9848	0.9777	0.9812
	Hybrid (Amplitude)	0.9878	0.9878	0.9885	0.9881

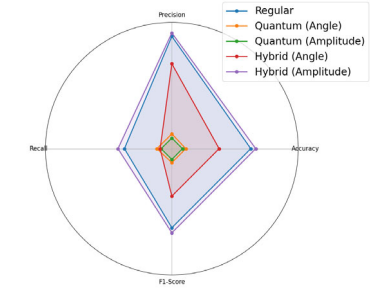
Table 3 Performance Comparison on ISCXIDS2012 Dataset

Model Type	Variant	Accuracy	Precision	Recall	F1-Score
SVM	Regular	0.9964	0.9973	0.9955	0.9964
	Quantum (Angle)	0.9488	0.9489	0.9488	0.9488
	Quantum (Amplitude)	0.9893	0.9893	0.9893	0.9893
	Hybrid (Angle)	0.9924	0.9937	0.9910	0.9924
	Hybrid (Amplitude)	0.9942	0.9964	0.9919	0.9942
KNN	Regular	0.9973	0.9982	0.9964	0.9973
	Quantum (Angle)	0.9946	0.9938	0.9956	0.9947
	Quantum (Amplitude)	0.9911	0.9947	0.9876	0.9911
	Hybrid (Angle)	0.9951	0.9955	0.9946	0.9951
	Hybrid (Amplitude)	0.9973	0.9982	0.9964	0.9973

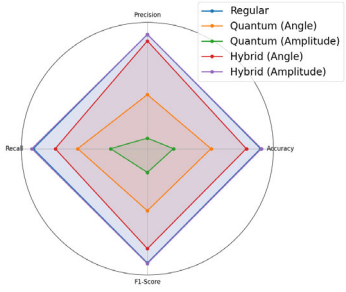
As indicated above, the performance of the proposed models was evaluated on two encrypted network traffic datasets: ISCX-URL2016 and ISCXIDS2012. Both these datasets contain flows labeled as benign or malicious, and are widely used for benchmarking in the encrypted traffic classification domain. Although these datasets are widely referenced, we acknowledge that they are relatively limited in diversity and size compared to more recent benchmarks such as CICIDS2017 or UNSW-NB15. As such, our evaluation should be regarded as a feasibility demonstration rather than a conclusive benchmark.

In the ISCX-URL-2016 dataset, the accuracy comparison in Fig. 5a shows that hybrid models, particularly Hybrid KNN, achieved performance comparable to the classical models. The differences are small and should be interpreted as maintaining strong baseline performance rather than establishing superiority. In contrast, models relying solely on quantum feature encoding, such as QSVM and QKNN, generally lagged behind the hybrid and classical models. This reinforces the practical value of hybrid approaches for current NISQ-era conditions. This is corroborated by the radar charts in Fig. 4a, where the hybrid variants display larger plots, indicating strong, balanced performance across all metrics and highlighting the potential of hybrid quantum-classical strategies for this task.

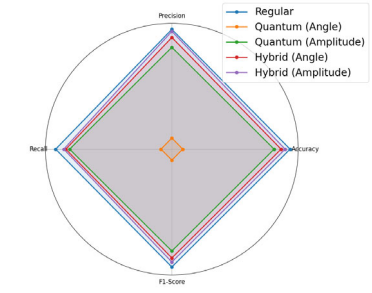
A different dynamic was observed for the ISCX-IDS-2012 dataset, where Fig. 5b shows exceptionally high accuracy for nearly all models. Here, the classical SVM and k-NN classifiers emerged as the top performers. While most quantum and hybrid models also achieved high scores, the QSVM with angle encoding showed lower relative performance. The radar charts in Fig. 4b confirm this; the classical SVM's plot encompasses the others, while the KNN variants all demonstrate remarkably consistent, high-level performance, indicating that the quantum and hybrid approaches successfully matched the excellent benchmark set by the classical algorithm on this dataset.



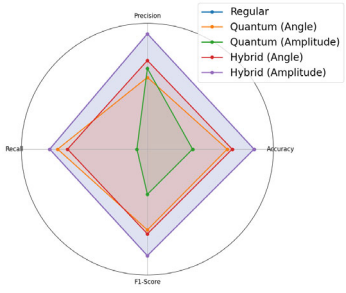
(a) Performance Metrics of SVM variants for ISCX-URL2016 Dataset



(b) Performance Metrics of KNN variants for ISCXIDS2016 Dataset

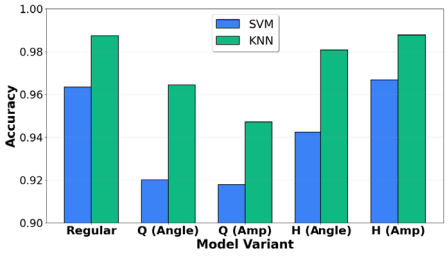


(c) Performance Metrics of SVM variants for ISCX-URL2012 Dataset

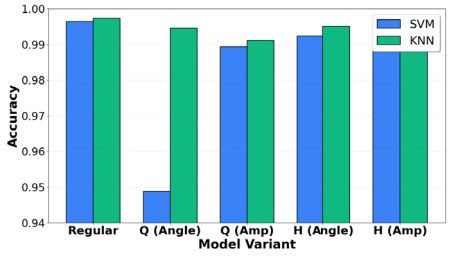


(d) Performance Metrics of KNN variants for ISCXIDS2012 Dataset

Figure 4 Radar charts comparing Accuracy, Precision, Recall, and F1-Score for SVM and KNN variants across the ISCX-URL2016 and ISCXIDS2012 datasets



(a) Accuracy Comparison for ISCX-URL2016 Dataset



(b) Accuracy Comparison for ISCXIDS2012 Dataset

Figure 5 Bar charts comparing accuracy of SVM and KNN variants across the ISCX-URL2016 and ISCXIDS2012 datasets. Q(Angle)-Quantum Angle Encoding, Q(Amp)- Quantum Amplitude Encoding, H(Angle)-Hybrid Angle Encoding, H(Amp)-Hybrid Amplitude Encoding

In both datasets, the KNN based methods outperformed the SVM based models. Though the quantum models did not achieve the highest overall accuracy, the results were competitive with the other models we used. This shows that quantum approaches have potential to be used in network traffic classification tasks as quantum hardware and algorithms continue to mature. The performance of the models was influenced by the type of encoding used. Amplitude encoding often yielded slightly stronger results than angle encoding, but this advantage was not consistent across all models or datasets, suggesting that encoding effectiveness may be context-dependent. These results suggest that amplitude encoding can be more effective for models like SVMs that rely on finding optimal hyperplanes that separate the data points. Unlike KNN, which depends on geometric distance metrics, SVM-based approaches can utilize better feature representation that can enhance the separation of classes. The hybrid models consistently outperformed their

purely quantum counterparts with both angle and amplitude encodings, while remaining comparable to classical baselines. This indicates that hybridization offers a balanced trade-off between leveraging quantum feature maps and maintaining robustness. The pure quantum approaches required substantially longer runtimes on simulators compared to classical and hybrid approaches. Hybrid models offered a more practical balance between execution cost and accuracy, making them more feasible for encrypted traffic analysis under current hardware limitations. All metrics were rounded to four decimal places.

6 Discussion and analysis

6.1 Performance of quantum models

The experimental results we obtained gave us some interesting insights into the capabilities of the machine learning models we used. The classical models gave a strong baseline performance for both the datasets. Both QSVM and QKNN performed inconsistently across the two datasets. On ISCX-URL2016, they lagged behind classical models, while on ISCXIDS2012 they approached, but did not surpass, classical baselines. As already indicated, this variability underscores the challenge of establishing consistent advantages for current fully quantum models under NISQ-era constraints. For instance, QSVM with angle encoding performed at approximately 0.9201, and with amplitude encoding slightly poorer at 0.9179. Similarly, QKNN achieved 0.9645 accuracy using angle encoding, outperforming QSVM but not the classical KNN which achieved 0.9874. Another observation was that angle encoding worked better than amplitude encoding for QKNN on this dataset, while the difference in performance between encodings was minimal for QSVM. These results show that the choice of encoding method can have a significant impact on model performance, potentially due to how different encodings represent feature relationships in the Hilbert space. On the ISCXIDS2012 Dataset, the quantum models performed much better. QSVM with amplitude encoding put up a good show with an accuracy of 0.9893, getting very close to the classical SVM which achieved 0.9964. QKNN also fared better here, with angle encoding scoring 0.9946 and amplitude encoding scoring about 0.9911. This is a considerable improvement from its performance on the ISCX-URL2016 dataset.

6.2 Feasibility of quantum models

An important aspect to look at is the practical feasibility of implementing the quantum algorithms. We conducted experiments on PennyLane-based quantum simulators, as real hardware was not available. However, simulators cannot fully capture hardware noise and fidelity constraints. Hence, while our results show feasibility, they may overestimate real-world performance. This limitation has been emphasized in prior foundational works (Schuld et al., 2014) and must be acknowledged explicitly. The quantum simulators were implemented using PennyLane, an open source Python framework. The pure quantum approaches required substantially longer runtime compared to classical and hybrid approaches, even on simulators. This highlights the scalability challenge of purely quantum models: runtime grows steeply with input dimension. Future work should quantify this cost for larger and more diverse encrypted traffic datasets. We were able to process feature vectors with dimensions suitable for our task on the quantum simulator. The pure quantum models i.e QSVM and QKNN take significantly more time for training and prediction than canonical ML models. This makes it computationally intensive to process

large datasets. This scalability issue is not as bad when it comes to the hybrid models. The hybrid models are significantly faster than the quantum models. They take a slightly longer time than the classical models because of the delay in computing the quantum encodings.

From a computational perspective, our current timing results reflect the cost of classical simulation of quantum circuits using PennyLane's lightning.qubit backend on CPU-only hardware. On both datasets, classical SVM and KNN complete training and prediction within seconds, whereas QSVM and QKNN are roughly one to two orders of magnitude slower due to repeated quantum kernel or fidelity evaluations on the simulator. The proposed hybrid models, which apply quantum encodings followed by classical SVM/KNN, narrow this gap and typically run within a small constant factor of the pure classical baselines, since the most expensive steps are handled by efficient scikit-learn routines. These measurements should therefore be interpreted as conservative, simulation-based upper bounds. A definitive assessment of runtime on actual quantum hardware will depend on future advances in qubit counts, coherence times, and noise levels that determine feasible circuit depths and sampling costs.

6.3 Hybrid models as intermediate solutions

When it comes to network traffic classification, hybrid models offer a good intermediate solution to improved classification performance. As discussed before, the hybrid KNN classifier with amplitude encoding performed on par or better than classical ML algorithms in most cases. The hybrid models can improve the strong baseline performance of classical models and are not as computationally intensive as pure quantum models. They provide us with an immediate, viable pathway to achieving superior performance. As real quantum hardware becomes readily available, pure quantum approaches can also be implemented for real life applications.

The work showcases the potential of Quantum Machine Learning (QML) in enhancing network security, with hybrid models offering a practical path toward industrial integration. The demonstrated success of the hybrid k-NN and SVM models, which outperformed or matched their classical counterparts in network traffic classification, highlights a key advantage for industrial applications: improved performance without the significant computational overhead of pure quantum systems. Industrial control systems (ICS) and operational technology (OT) environments are increasingly targeted by sophisticated cyber threats. The integration of validated hybrid QML models may strengthen cybersecurity in industrial systems by offering performance comparable to classical baselines while enabling experimentation with quantum encodings. However, our results do not demonstrate clear superiority over advanced deep learning approaches (e.g., transformer-based anomaly detectors). Future comparative studies with such baselines are essential for fair benchmarking. The ability of these hybrid models to balance performance and execution time makes them an attractive, immediate solution for industrial settings where uptime and low latency are critical. As quantum hardware continues to mature, the groundwork laid by these hybrid approaches will pave the way for the eventual deployment of more powerful, pure quantum security solutions, ensuring a scalable and future-proof strategy for protecting vital industrial infrastructure.

During implementation, the main computational bottleneck arose from repeatedly evaluating quantum circuits to compute the Gram matrix for QSVM and the pairwise fidelities for QKNN on the statevector simulator. In the initial, non-parallelized version, these

kernel and distance computations were executed sequentially over all training–test pairs, leading to long wall-clock times and poor CPU utilisation. By parallelising these evaluations with joblib (using multiple cores for independent circuit runs), and by exploiting the symmetry of the Gram matrix to compute only its lower triangle, we substantially reduced runtime. In our experiments, this optimisation yielded practical speedups depending on dataset size turning QSVM/QKNN training and prediction from prohibitively slow into manageable, albeit still costlier than purely classical baselines.

Our work directly addresses the critical challenges involved in analyzing the encrypted network traffic. Our work has explored how QML models — individually and in combination with classical approaches — can be leveraged to classify and analyze encrypted network traffic without decrypting it. This enables non-intrusive, privacy-preserving security monitoring, which is crucial for industrial environments where downtime, data breaches, or operational disruptions can have large-scale impacts. The integration of quantum-enhanced models into industrial information systems contributes to real-time anomaly detection, secure data flow governance, and adaptive cybersecurity analytics in industrial control systems (ICS), smart manufacturing platforms, and edge-cloud integrated infrastructures. Furthermore, our comparative analysis supports the design of next-generation AI-based network security modules that can be embedded within industrial data integration pipelines, fostering resilient, secure, and intelligent industrial ecosystems.

While our study provides insights into the use of QML for encrypted traffic analysis, several limitations remain. First, our reliance on simulators means hardware-specific effects (e.g., gate noise, decoherence) were not captured. Second, scalability remains an open issue: runtime and memory requirements of amplitude encoding increase exponentially with feature dimensions. Third, our evaluation used only two datasets of moderate size; broader validation across modern benchmarks such as CICIDS2017 is needed. Finally, we did not compare against deep learning baselines such as CNN- or transformer-based anomaly detectors, which currently represent the state of the art in traffic classification (Wang et al., 2021; Lotfollahi et al., 2020). These constraints frame our results as preliminary and motivate future work.

It is natural that more sophisticated deep learning architectures, could narrow or even invert some of the performance gaps observed between classical ML and hybrid QML models. At the same time, recent comparative works also show that classical methods can remain competitive with deep learning in several traffic-analysis scenarios, especially when feature engineering is strong and datasets are structured. Against this backdrop, our present results should be interpreted as a comparison within a "traditional ML + (pure/hybrid) QML" design space [21].

7 Conclusion and future work

In this study, we conducted a comprehensive investigation into the application of quantum and hybrid quantum-classical machine learning models for encrypted network traffic analysis. Using two benchmark datasets from the Canadian Institute of Cybersecurity, we first preprocessed the encrypted traffic data and then applied quantum feature encoding techniques—specifically angle encoding and amplitude encoding—to represent the data in quantum-compatible formats. We developed and evaluated three types of models: purely classical machine learning models such as K-Nearest Neighbors (KNN) and Support Vector Machines (SVM); fully quantum models including Quantum KNN (QKNN)

and Quantum SVM (QSVM); and hybrid models that combined quantum encodings with classical classifiers. The performance of these models was rigorously assessed using standard metrics such as accuracy, precision, recall, and F1-score. Our analysis revealed that hybrid models, particularly those utilizing amplitude encoding with classical classifiers like KNN and SVM, often performed on par with or better than their purely classical counterparts. While fully quantum models showed promise, their performance was generally constrained by current quantum hardware and simulator limitations. In the present study, the accuracies reported in Tables 1 and 2 are point estimates obtained from a single train–test split for each configuration. We acknowledge that best practice would be to complement these values with estimates of variability (e.g., standard deviations or confidence intervals) obtained from repeated runs or cross-validation. Given the relatively large test sets and the magnitude of the observed differences, we expect the main qualitative trends, namely that the hybrid QML models are competitive with or slightly superior to their purely classical counterparts on the considered datasets, to remain consistent. However, a full statistical characterisation of uncertainty is left as future work. This work demonstrates the practical potential of hybrid quantum approaches for analyzing encrypted traffic and highlights their relevance to industrial information integration, especially in cybersecurity applications that require handling of privacy-preserving encrypted data.. Our experiments were conducted using quantum simulators implemented with the help of PennyLane. Future research can focus on testing these models on real quantum hardware. The empirical evaluation is restricted to two CIC datasets, ISCX-URL2016 and ISCX-IDS2012. Both are widely used and well-documented benchmarks for encrypted traffic classification. Given the substantial simulation cost of QSVM and QKNN on CPU-only hardware using PennyLane, we chose to focus on these two datasets as a controlled feasibility study of quantum and hybrid QML models for ENTA, rather than a comprehensive benchmark. Extending the analysis to newer and more heterogeneous datasets is an important direction for future work to further assess the robustness and generality of the observed trends. Our work shows that quantum enhanced machine learning holds great promise for encrypted network traffic classification. Such quantum enhanced models can be applied to other fields as well.

Different quantum encoding schemes offer complementary strengths for encrypted network traffic metadata. Basis (computational) encoding aligns naturally with categorical or discrete protocol information (e.g., flags, ports, protocol IDs) and hierarchical protocol fields, though it scales linearly in qubit count and does not by itself induce rich nonlinear feature maps. Rotation-based encodings (such as our angle encoding) are convenient for continuous features like flow duration, packet sizes, and inter-arrival times, and can also accommodate mixed continuous–categorical data after suitable preprocessing, while remaining relatively shallow and hardware-friendly. Hamiltonian and IQP-style encodings have the potential to capture more complex interactions across features and protocol layers by embedding data into structured phase evolutions, at the cost of deeper and less interpretable circuits. Finally, quantum convolutional encodings and QCNN architectures, with their local receptive fields and pooling operations, appear particularly promising for modelling temporal dependencies and hierarchical organisation in packet- and flow-level representations of encrypted traffic. This is a promising topic of future research.

From a deployment perspective, hybrid QML is most naturally integrated as an enhancement to existing IDS and ENTA pipelines. Quantum encoders act as a feature-enrichment layer and the downstream decision logic remains purely classical. In a typical industrial setting, flow metadata (e.g., 5-tuple, packet sizes, inter-arrival times, protocol flags) would first be aggregated by existing collectors, then periodically batch-encoded via quantum circuits (on simulators in the near term, and on dedicated QPUs in the longer term), producing compact quantum-enhanced feature vectors that feed standard SVM/KNN-style detectors or more advanced SOC analytics. Such a design allows operators to reuse current data ingestion, alerting, and incident-response infrastructure while gradually offloading only the encoding step to quantum resources, making hybrid QML a realistic candidate for incremental adoption in Security Operation Centers (SOC), Software Defined Network (SDN) middleboxes, or monitoring gateways as quantum hardware matures.

Appendix A: Quantum circuits

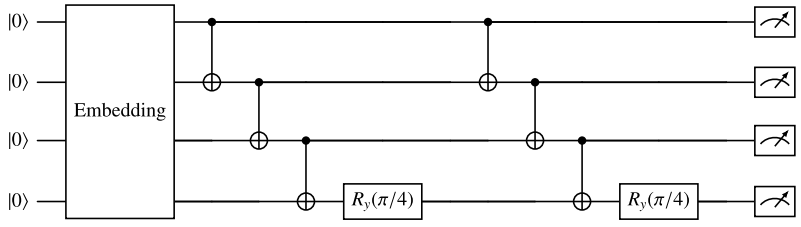


Figure 6 **QKNN**: Quantum K-Nearest Neighbors circuit structure

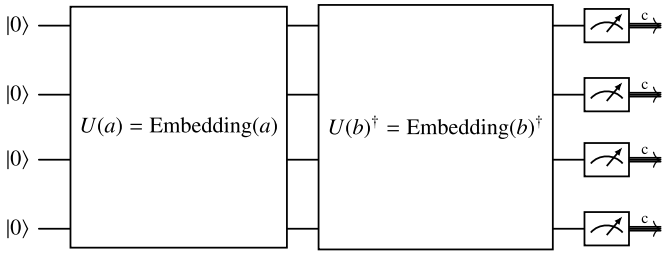


Figure 7 **QSVM**: Quantum Support Vector Machine circuit for kernel evaluation

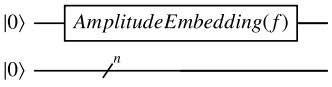


Figure 8 Hybrid Quantum Machine Learning (QML) Circuit: A classical data vector f is embedded into a quantum state via amplitude embedding across n qubits

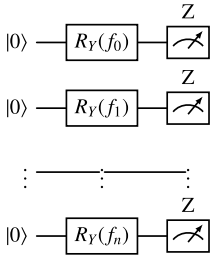


Figure 9 Hybrid Quantum Machine Learning (QML) circuit: Classical input features $f_0, f_1, \ldots, f_n$ are encoded into quantum states using Y-axis rotations via AngleEmbedding. Each qubit is then measured in the Z-basis to extract quantum-transformed features

Appendix B: Sample quantum encodings

In this section, we present examples for two quantum data encoding techniques: *angle encoding* and *amplitude encoding*.

B.1 Angle encoding example

Angle encoding maps classical data values to rotation angles on qubits. A common approach is to use $R_Y(\theta)$ rotations, where each data value determines the rotation angle θ.

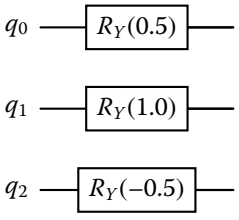
Consider the classical vector:

$$\mathbf{x} = [0.5, 1.0, -0.5]$$

These values are directly mapped to rotations around the Y-axis. The resulting quantum state is:

$$|\psi\rangle = R_Y(0.5)|0\rangle \otimes R_Y(1.0)|0\rangle \otimes R_Y(-0.5)|0\rangle$$

We can visualise this with the following quantum circuit:



Each qubit starts in the $|0\rangle$ state and is rotated based on the corresponding value from the input vector.

B.2 Amplitude encoding example

Amplitude encoding encodes a classical vector into the amplitudes of a quantum state. For the vector:

$$\mathbf{x} = [1, 2, 0, 1]$$

We first normalize it:

$$||\mathbf{x}|| = \sqrt{1^2 + 2^2 + 0^2 + 1^2} = \sqrt{6} \Rightarrow \mathbf{x}_{\text{norm}} = \left[\frac{1}{\sqrt{6}}, \frac{2}{\sqrt{6}}, 0, \frac{1}{\sqrt{6}}\right]$$

Then we get the amplitude encoded quantum state:

$$|\psi\rangle = \frac{1}{\sqrt{6}}|00\rangle + \frac{2}{\sqrt{6}}|01\rangle + 0 \cdot |10\rangle + \frac{1}{\sqrt{6}}|11\rangle$$

This is how amplitude encoding is done.

Appendix C: Hyperparameter evaluation

We have documented the different performance metrics obtained for the different hyperparameter configurations. The results are presented in the table below.

Table 4 Performance comparison of QSVM/QKNN and Classical/Hybrid SVM/KNN models on ISCX-URL2016 and ISCXIDS2012 datasets for different hyperparameters

Section	Model Type	Variant	Accuracy	Precision	Recall	F1-Score
(a)						
QSVM and QKNN on ISCX-URL2016	QSVM	γ='auto', shrinking=False (Angle)	0.8933	0.9366	0.8526	0.8926
		γ='scale', shrinking=True (Angle)	0.8967	0.9110	0.8808	0.8956
		γ='auto', shrinking=False (Amplitude)	0.8967	0.9110	0.8808	0.8956
		γ='scale', shrinking=True (Amplitude)	0.8867	0.9091	0.8609	0.8844
	QKNN	n=5 (Angle)	0.9233	0.8854	0.9653	0.9236
		n=15 (Angle)	0.8967	0.8424	0.9653	0.8997
		n=5 (Amplitude)	0.8233	0.8013	0.8403	0.8203
		n=15 (Amplitude)	0.7700	0.7820	0.7222	0.7509
(b)						
QSVM and QKNN on ISCXIDS2012	QSVM	γ='scale', shrinking=True (Angle)	0.9667	0.9504	0.9829	0.9664
		γ='auto', shrinking=False (Angle)	0.9458	0.9048	0.9913	0.9461
		γ='scale', shrinking=True (Amplitude)	0.9750	0.9580	0.9913	0.9744
		γ='auto', shrinking=False (Amplitude)	0.9458	0.9048	0.9913	0.9461
	QKNN	n=5 (Angle)	0.9708	0.9833	0.9594	0.9712
		n=15 (Angle)	0.9542	0.9828	0.9268	0.9540
		n=5 (Amplitude)	0.9667	0.9752	0.9594	0.9672
		n=15 (Amplitude)	0.8750	0.8550	0.9106	0.8819
(c)						
Classical/Hybrid SVM & KNN on ISCX-URL2016	SVM	γ='scale', rbf kernel (Amplitude)	0.9586	0.9740	0.9445	0.9590
		γ='scale', rbf kernel (Angle)	0.9423	0.9674	0.9186	0.9424
		γ='auto', poly kernel, degree=3 (Angle)	0.9005	0.9354	0.8660	0.8994
		γ='scale', rbf kernel (Classical)	0.9582	0.9790	0.9388	0.9584
		γ='auto', poly kernel, degree=3 (Classical)	0.8680	0.9831	0.7558	0.8546
	KNN	n=5, minkowski, uniform (Amplitude)	0.9837	0.9800	0.9885	0.9842
		n=7, euclidean, distance (Amplitude)	0.9878	0.9878	0.9885	0.9881
		n=5, minkowski, uniform (Angle)	0.9763	0.9763	0.9777	0.9770
		n=7, euclidean, distance (Angle)	0.9808	0.9848	0.9777	0.9812
		n=5, minkowski, uniform (Classical)	0.9815	0.9785	0.9856	0.9821
		n=7, euclidean, distance (Classical)	0.9874	0.9878	0.9878	0.9878
(d)						
Classical/Hybrid SVM & KNN on ISCXIDS2012	SVM	γ='scale', rbf kernel (Amplitude)	0.9942	0.9964	0.9919	0.9942
		γ='scale', rbf kernel (Angle)	0.9760	0.9935	0.9579	0.9754
		γ='auto', poly kernel, degree=3 (Angle)	0.8851	0.8588	0.9203	0.8885
		γ='scale', rbf kernel (Classical)	0.9960	0.9973	0.9946	0.9960
		γ='auto', poly kernel, degree=3 (Classical)	0.9328	0.8833	0.9964	0.9365
	KNN	n=5, minkowski, uniform (Amplitude)	0.9969	0.9982	0.9955	0.9969
		n=7, euclidean, distance (Amplitude)	0.9969	0.9982	0.9955	0.9969
		n=5, minkowski, uniform (Angle)	0.9942	0.9964	0.9919	0.9942
		n=7, euclidean, distance (Angle)	0.9947	0.9982	0.9910	0.9946
		n=5, minkowski, uniform (Classical)	0.9964	0.9973	0.9955	0.9964
		n=7, euclidean, distance (Classical)	0.9969	0.9982	0.9955	0.9969

Author contributions
G.S: Data curation, software, Investigation,writing A.M:Data curation, software, Investigation,writing A.P: Formal Analysis, Writing & Reviewing, editing, Project administration A.K: Conceptualization & Methodology

Funding information
Open access funding provided by Vellore Institute of Technology.

Data availability
No datasets were generated or analysed during the current study.

Declarations

Use of generative AI
Generative AI tools were used for grammar suggestions and formatting improvements in this paper. All content and analysis in this paper were authored and reviewed by the authors.

Competing interests
The authors declare no competing interests.

Received: 24 September 2025 Accepted: 19 December 2025 Published online: 20 January 2026

References

1. Afham A, Basheer A, Goyal S. Quantum k-nearest neighbor machine learning algorithm. arXiv preprint. arXiv:2003.09187 (2020). https://doi.org/10.48550/arXiv.2003.09187.
2. Alwhbi IA, Zou CC, Alharbi RN. Encrypted network traffic analysis and classification utilizing machine learning. Sensors. 2024;24:3509. https://doi.org/10.3390/s24113509.
3. Asadizanjani N, Reddy Kottur H, Dalir H. Quantum computing, wearables, and next-gen ic packaging. In: Introduction to microelectronics advanced packaging assurance. Springer; 2025. p. 161–80.
4. Bazammul A, Karkheiran K, Jahannia B, Hayden B, Amirany A, Patil C, Heidari E, Dalir H. High-speed dac/adc implementation using rfsoc fpga for quantum computing. In: Photonic computing: from materials and devices to systems and applications II. SPIE; 2025. p. 42–50.
5. Bhaskaran P, Prasanna S. An accuracy analysis of classical and quantum machine learning algorithms using different datasets. In: 2024 2nd international conference on networking and communications (ICNWC). 2024. p. 1–8. https://doi.org/10.1109/ICNWC60771.2024.10537433.
6. Canadian Institute for Cybersecurity. ISCX IDS 2012 intrusion detection evaluation dataset. 2012. https://www.unb.ca/cic/datasets/ids.html.
7. Canadian Institute for Cybersecurity. ISCX-URL2016 dataset. 2016. https://www.unb.ca/cic/datasets/url-2016.html.
8. Chae E, Choi J, Kim J. An elementary review on basic principles and developments of qubits for quantum computing. Nano Converg. 2024;11:11. https://doi.org/10.1186/s40580-024-00418-5.
9. Cherukuri A, Ikram S, Li G, Liu X. Encrypted Network Traffic Analysis. SpringerBriefs in Computer Science. Springer International Publishing. 2024. https://books.google.co.in/books?id=c9sVEQAAQBAJ.
10. Cui R, Lyu Z. Analysis of quantum gates in quantum circuits. Theor Nat Sci. 2023;10:1–8. https://doi.org/10.54254/2753-8818/10/20230301.
11. Darwish S, Ali R, Elzoghabi A. Quantum-inspired k-nearest neighbors classifier for enhanced printer source identification in forensic document analysis. Sci Rep. 2025;15:4097. https://doi.org/10.1038/s41598-025-86558-y.
12. Elmaghraby R, Aziem N, Sobh M, Bahaa-Eldin A. Encrypted network traffic classification based on machine learning. Ain Shams Eng J. 2024;15:102361. https://doi.org/10.1016/j.asej.2023.102361.
13. Guerrero-Estrada AY, Quezada LF, Sun GH. Benchmarking quantum versions of the knn algorithm with a metric based on amplitude-encoded features. Sci Rep. 2024;14. https://doi.org/10.1038/s41598-024-67392-0.
14. Hdaib M, Rajasegarar S, Pan L. Quantum deep learning-based anomaly detection for enhanced network security. Quantum Mach Intell. 2024;6:16. https://doi.org/10.1007/s42484-024-00163-2.
15. He Q, Yang W, Bingren C, Geng Y, Huang L. Transnet: training privacy-preserving neural network over transformed layer. Proc VLDB Endow. 2020;13:1849–62. https://doi.org/10.14778/3407790.3407794.
16. Jain S, Gandhi A, Singla S, Garg L, Mehla S. Quantum machine learning and quantum communication networks: the 2030s and the future. In: 2022 international conference on computational modelling, simulation and optimization (ICCMSO). 2022. p. 59–66. https://doi.org/10.1109/ICCMSO58359.2022.00025.
17. Jhanwar A, Nene MJ. Enhanced machine learning using quantum computing. In: 2021 second international conference on electronics and sustainable communication systems (ICESC). 2021. p. 1407–13. https://doi.org/10.1109/ICESC51422.2021.9532638.
18. Jin Z, Duan K, Chen C, He M, Jiang S, Xue H. Fedetc: encrypted traffic classification based on federated learning. Heliyon. 2024;10:e35962. https://doi.org/10.1016/j.heliyon.2024.e35962.
19. Joseph S, Joshi H, Hassan M, Bairagi A. Advancing quantum machine learning: from theoretical concepts to experimental implementations. 2024. https://doi.org/10.2139/ssrn.4946682.
20. Kalinin M, Krundyshev V. Analysis of a huge amount of network traffic based on quantum machine learning. Autom Control Comput Sci. 2021;55:1165–74. https://doi.org/10.3103/S014641162108040X.
21. Lichy A, Bader O, Dubin R, Dvir A, Hajaj C. When a rf beats a cnn and gru, together—a comparison of deep learning and classical machine learning approaches for encrypted malware traffic classification. Comput Secur. 2023;124:103000.
22. Lotfollahi M, Shirali Hossein Zade R, Jafari Siavoshani M, Saberian M. Deep packet: a novel approach for encrypted traffic classification using deep learning. Soft Comput. 2020;24:1999–2012. https://doi.org/10.1007/s00500-019-04030-2.

23. Lu W, Lu Y, Li J, Sigov A, Ratkin L, Ivanov LA. Quantum machine learning: classifications, challenges, and solutions. J Ind Inf Integr. 2024;42:100736. https://doi.org/10.1016/j.jii.2024.100736.
24. Medeiros de Abreu D, Moura D, Esteve Rothenberg C, Abelém A. Quantumnetsec: quantum machine learning for network security. Int J Netw Manag. 2025;35:e70018. https://doi.org/10.1002/nem.70018.
25. Park JE, Quanz B, Wood S, Higgins H, Harishankar R. Practical application improvement to quantum svm: theory to practice. 2020. https://doi.org/10.48550/arXiv.2012.07725.
26. Rath M, Date H. Quantum data encoding: a comparative analysis of classical-to-quantum mapping techniques and their impact on machine learning accuracy. EPJ Quantum Technol. 2024;11:72. https://doi.org/10.1140/epjqt/s40507-024-00285-3.
27. Rezaei S, Liu X. Deep learning for encrypted traffic classification: an overview. IEEE Commun Mag. 2019;57:76–81. https://doi.org/10.1109/MCOM.2019.1800819.
28. Schuld M, Sinayskiy I, Petruccione F. An introduction to quantum machine learning. Contemp Phys. 2014;56:172–85. https://doi.org/10.1080/00107514.2014.964942.
29. Shen M, Ye K, Liu X, Zhu L, Kang J, Yu S, Li Q, Xu K. Machine learning-powered encrypted network traffic analysis: a comprehensive survey. IEEE Commun Surv Tutor. 2022;25:791–824.
30. Wang Z, Fok KW, Thing V. Machine learning for encrypted malicious traffic detection: approaches, datasets and comparative study. Comput Secur. 2021;113:102542. https://doi.org/10.1016/j.cose.2021.102542.
31. Wu T, Jahannia B, Ye J, Cai Q, Amirany A, Asadizanjani N, Dalir H, Heidari E. Quantum integrated squeezed light enhancement and detection with photonics integrated circuits: a single-pass configuration. In: Quantum communications and quantum imaging XXIII. SPIE; 2025. p. 173–83.
32. Yin T. Quantum support vector machines: theory and applications. Theor Nat Sci. 2024;51:34–42. https://doi.org/10.54254/2753-8818/51/2024CH0158.
33. Zardini E, Blanzieri E, Pastorello D. A quantum k-nearest neighbors algorithm based on the Euclidean distance estimation. Quantum Mach Intell. 2024;6:23. https://doi.org/10.1007/s42484-024-00155-2.

Publisher's note