Chapter 7

Projected Quantum Kernel based Link Prediction in Dynamic Networks (PQKLP)

In this chapter, A Projected Quantum Kernel-based Link Prediction (PQKLP) [1], a feature-based framework for quantum-enhanced link prediction problems in dynamic networks is introduced. It combines the disciplines of social networks and quantum computing. It employed high-dimensional Hilbert spaces to enhance the prediction data in this model, which otherwise we only have access to via inner products provided by measurements. Such enhancement leads to better prediction results from machine learning-based link prediction techniques. We trained six classical machine learning models and their quantum-enhanced counterparts based on the enhanced features generated by the Projected Quantum Kernel (PQK) technique.

[1] Published in Journal Computer Communications, PQKLP: Projected Quantum Kernel based Link prediction in Dynamic Networks

7.1 Introduction

Quantum computing [188–190] is a new branch of computer science that uses quantum mechanics [191] to solve problems. Quantum theory has been the most thorough account of small scale physics since its birth at the turn of the twentieth century. Quantum Machine Learning (QML) is a multidisciplinary field that brings together two of the most fascinating research areas of quantum computing and classical machine learning. Quantum computing and machine learning are undeniably "hot" issues in research and industry. QML models have been shown to have a mathematical structure that is quite similar to kernel methods, i.e., they analyses data in high-dimensional Hilbert spaces to which we only have access through inner products revealed by measurements [43]. Machine learning is one of the most intriguing possible uses of quantum computing.

Quantum systems are distinguished by a generalization of probability theory that allows for unique phenomena such as superposition and entanglement that are impossible to simulate with a standard computer [42]. Quantum computers can perform rapid linear algebra on a state space that expands exponentially with the number of "qubits". This is one of the main breakthroughs that has led to their use in machine learning. These quantum accelerated linear-algebra based machine learning algorithms include principal component analysis [192], support vector machine [193], K-means clustering [194], and recommendation systems [195]. From the link prediction research perspective, many supervised machine learning (ML) classification techniques may be used to address the link prediction problem [1, 50, 196]. Multiple studies have shown that this method delivers good results; nevertheless, choosing the collection of features (variables) to train classifiers remains a key challenge.

In this study, we provide a solution to the issue of link prediction in dynamic networks using supervised learning and the Projected Quantum Kernel (PQK). Integration of several types of structural data from snapshots is the driving force behind our proposed technique. The proposed technique utilizes feature vectors of node pairs to account for different types of structural topological data from all snapshots. We analysed data in high-dimensional Hilbert spaces using a number of well-known similarity indices as features, including Common Neighbors (CN), Adamic/Adar Index (AA), Jaccard Coefficient (JC), Preferential Attachment (PA), and Shortest Path index (SP). Due to developments in computer power and algorithmic innovation, machine learning methods have become effective tools for spotting patterns in data. It is conceivable to expect that quantum computers may outperform classical computers in machine learning tasks since quantum systems display aberrant behavior that conventional systems are thought to be incapable of creating. Quantum machine learning is the study of how to create and implement quantum algorithms to allow machine learning on conventional computers that is more accurate.

The following are the primary motivations for developing this approach.

- Machine learning algorithms that naturally rely on quantum features to increase their performance have attracted much interest. Hsin-Yuan Huang et al. [3] have shown that quantum machine learning outperforms classical machine learning in some cases. Based on this assumption, in this research, we expanded the concept of quantum learning to predict the future or missing link in a temporal network.
- Quantum models have a mathematical framework that is quite similar to kernel methods, i.e., they evaluate data in high-dimensional Hilbert spaces to which we can only acquire access via inner products disclosed by measurements [43]. Kernel-based methods have been employed with good results in link prediction recently [44–46].
- By projecting back from the quantum space to a classical one in the projected quantum kernel model [3], it is our belief that the underlying patterns in data can be enhanced.

The main contribution of this work are as follows.

- In this paper we have formalized the use of QML techniques to solve the problem of link prediction in social networks. Using the Projected Quantum Kernel and machine learning models, we present a novel approach, PQKLP, for solving the link prediction problem employing both local and global information. To the best of our knowledge this is the first attempt to solve the link prediction problem in dynamic networks using PQK enhanced techniques.

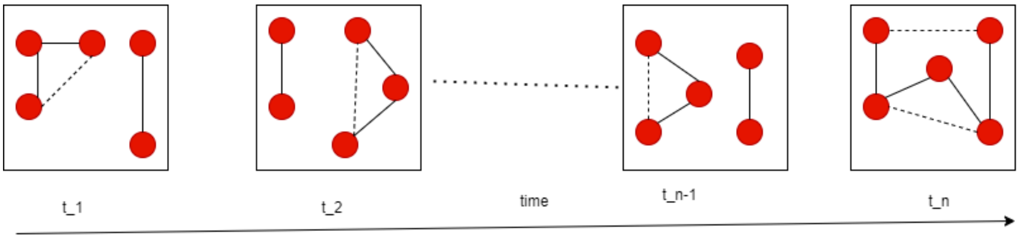
- Using PQK we have transformed the popularly used snapshot-based feature set form into quantum space such that the accuracy of machine learning-based link prediction can be improved.

- Using this PQK transformed feature set, we have demonstrated the relative superiority of our approach from others which contain even higher number of individual features. This shows that the proposed transformation enhances data patterns in such a way so as to make the task of machine learning more accurate.

- We have compared the results of our proposed approach i.e Quantum enhanced neural Network (PKLPQ-NN), Quantum enhanced XGBoost (PQKLP-XGB), Quantum enhanced Logistic Regression (PKLPQ-LR), Quantum Random forest classifier (PQKLP-RFC), Quantum Linear discriminant Analysis (PQKLP-LDA) and Quantum Gaussian Naive Bayes (PQKLP-GNB) with corresponding machine learning models Neural Network (NN), XGBoost (XGB), Logistic Regression (LR), Random forest classifier (RFC), Linear discriminant Analysis (LDA) and Gaussian Naive Bayes (GNB). Extensive research on five well-known dynamic datasets with four performance evaluation matrices demonstrates that the proposed approach produces enhanced results in cases of PQKLP-RFC, PQKLP-XGB and PQKLP-NN. These approaches even outperform four state-of-the-art algorithms.

- The QML approaches use the huge dimensionality of quantum Hilbert space to get an optimized solution by modeling the feature space of a classification problem with a quantum state. To address this issue we have explored supercomputer based implementation solutions in this paper.

FIGURE 7.1: Temporal Networks



7.1.1 Quantum Computation

Before we go into our approach, let's go through the basics of quantum computing (such as Qubits, Hilbert spaces and circuital implementation using Cirq):

- **The Qubit.** In a classical computer, a bit is the smallest unit of information. A quantum bit, also known as a qubit, is the smallest unit of information held in a two-state quantum computer [188, 191, 197]. A qubit is a quantum mechanical system with two levels. To show it, quantum states are employed. A qubit is a quantum particle with two distinct states that can be measured. A qubit differs from a conventional bit in that a bit in a classical system can only have one of two values: 0 or 1, whereas a qubit can have any value between 0 and 1, signifying superposition of states. It is common practice in quantum physics to refer to elements ψ of abstract complex vector spaces as $|\psi\rangle$ kets rather than vectors, and to use vertical bars and angular brackets to symbolize them. It is represented by by a pair of complex numbers (α, β).

  State of a Qubit. The state of qubit is a vector represented as$|\psi(t)|$. It has the information about system at a particular given time. It is a member of Hilbert Space and is dynamic in nature. Mathematically, the state of qubit, ψ is represented as.

$$|\psi\rangle = \alpha|0\rangle + \beta|0\rangle \tag{7.1}$$

The probability amplitudes of the 0 and 1 states are represented by the complex numbers α, and β respectively. The complex numbers satisfy the following condition.

$$\alpha^2 + \beta^2 = 1 \tag{7.2}$$

Here, α^2 and β^2 are probabilities of qubit in 0 and 1 state. In two-dimensional complex vector space, a qubit's quantum state is represented by a unit-length vector in state space.

- **The Hilbert Space.** The Hilbert Space(H) is a special kind of linear vector space. It has all the properties of linear vector space with some additional properties .

  Properties of Hilbert Space.

  - Hilbert Space has an inner product operation which satisfy certain condition. This Inner product can be defined as: Let $\langle \psi_1, \psi_2 \rangle \in$ set of complex numbers.

    $$Inner\_product\_of\_vectors = \psi_1 \cdot \psi_2 \tag{7.3}$$

    Constraints of Inner product.

    1. Conjugate property.

       $$\langle \psi_1, \psi_2 \rangle = \langle \psi_2, \psi_1 \rangle^* \tag{7.4}$$

       Inner product of two elements in Hilbert space is complex conjugate of the inner product of two elements in opposite order.

    2. Linear with respect to second vector

       $$\langle \psi_1, a\psi_2 + b\psi_3 \rangle = a\langle \psi_1, \psi_2 \rangle + b\langle \psi_1, \psi_3 \rangle \tag{7.5}$$

    3. Antilinear with respect to first vector

       $$\langle a\psi_1 + b\psi_2, \psi_3 \rangle = a^*\langle \psi_1, \psi_3 \rangle + b^*\langle \psi_2, \psi_3 \rangle \tag{7.6}$$

    4. Inner product of a vector with itself must not be negative.

       $$\langle \psi, \psi \rangle = |\psi|^2 \geq 0 \tag{7.7}$$

It is zero when the vector itself is zero. It is called positive definiteness.

5. Distance between vectors in Hilbert space

$$|\langle \psi_1 - \psi_2 \rangle| = \sqrt{\langle \psi_2 - \psi_1, \psi_2 - \psi_1 \rangle} = d \tag{7.8}$$

  - Hilbert spaces are separable. They contain a countable, dense subset.

$$S = \{\phi_n\} \tag{7.9}$$

  - Helbert spaces are complete (no gaps).

$$\lim_{m,n \to \infty} |\psi_1 - \psi_2| = 0 \tag{7.10}$$

$$\lim_{n \to \infty} |\phi - \psi_n| = 0 \tag{7.11}$$

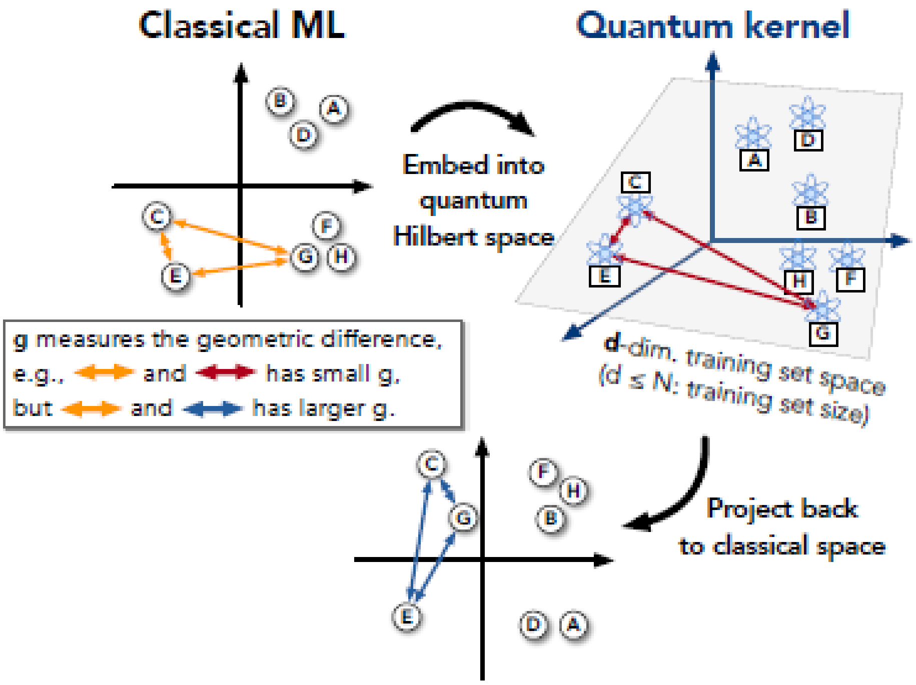
Here ϕ is an element in Hilbert space.

- **Cirq.** Cirq [198] is a free software framework for triggering quantum circuits. It includes the fundamental structures required for describing quantum computations, such as qubits, gates, circuits, and measurement operators. Quantum calculations defined by the user can subsequently be run in a virtual environment or on real hardware. Cirq also includes tools like compilers and schedulers that assist users in creating efficient NISQ algorithms.

7.1.2 Projected Quantum Kernels (PQK)

PQK kernels [3] function by projecting quantum states into a representation that is mostly classical, for as by using reduced physical observable or classical shadows [199–203]. Even if the training set space has an enormous dimension, projection enables us to reduce

FIGURE 7.2: Projected Quantum Kernel [3]


Classical ML
Quantum kernel
B A D
C F G H E
Embed into quantum Hilbert space
A D C B E H F G
g measures the geometric difference, e.g., ⟷ and ⟷ has small g, but ⟷ and ⟷ has larger g.
d-dim. training set space (d ≤ N: training set size)
Project back to classical space
C F H G B E D A

the original feature set to a low-dimensional classical space with enhanced generalization features. Since it traverses the exponentially vast quantum Hilbert space, the projected quantum kernel may be difficult to analyze without a quantum computer. In numerical testing, we observed that for traditional Machine Learning-based (ML) prediction models, the classical projection actually increases rather than decreases the geometric distance. A simple quantum model can learn exponentially more samples using the quantum kernel $(Tr(\rho(x_i)\rho(x_j)))$ but only a linear number of samples using a classical ML model. The one-particle reduced density matrix (1-RDM) measurement on all qubits for the encoded state is one of the simplest types of projected quantum kernel. Based on assumption that $\rho_k(x_i) = Tr_{j\neq k}[\rho(x_i)]$, this kernel [3] is defined as

$$k^{PQ}(x_i, x_j) = exp\left(-\gamma\sum_k \|\rho_k(x_i) - \rho_k(x_j)\|_F^2\right) \tag{7.12}$$

This kernel provides a 1-RDM feature map function that may express arbitrary functions of quantum state 1-RDM powers. According to the non-intuitive implications of density functional theory, even one-body densities may be sufficient for determining the exact ground state [204] and time-dependent density characteristics of many-body systems given reasonable assumptions [205]. This provides a method for efficiently building a kernel function with all orders of RDMs using local randomised measurements and the traditional shadows formalism [200]. The typical shadow formalism permits the quick construction of RDMs from a minimal quantity of data. In a newly suggested learning problem based on discrete logarithms [206], projected versions of quantum kernels provide a straightforward and rigorous quantum speedup.

The conventional ML models and quantum-enhanced ML models are depicted in Figure 7.2. The data points A, B, C,...,etc., are in different spaces. An arrow represents the kernel function, which is a measure of data similarity. The effective dimension of the data set is d, and the geometric difference between similarity measures in different machine learning models is g in the quantum Hilbert space. The quantum Hilbert space is utilized to define the kernel function that will be used to train the model in the quantum kernel technique.

Hsin-Yuan Huang et al. [3] has shown the advantage of quantum-enhanced machine learning over classical machine learning and vice-versa. In this work we have use the concept of quantum learning and extended the idea of quantum learning to predict future or missing link in dynamic network. The basic code for quantum transformation of data is available as an open source [2]. The formulation of model using Projected Quantum Kernel and Link prediction is shown in Fig. 7.3.

The performance of the machine learning-based prediction approaches can be significantly enhanced by projecting back from the quantum space to a classical one in the projected quantum kernel model, according to [3]. When the complete exponential quantum state space is used to generate the kernel function, $k(x_i,x_j) = Tr(\rho(x_i)\rho(x_j))$, we have observed that employing the native quantum state space to define the kernel function can fail to learn even a basic function. Otherwise, the quantum machine

[2] https://www.tensorflow.org/quantum/tutorials/quantum_data/

learning model may be reproduced traditionally and a significant advantage would be lost [3]. *RDM*s are defined in a classical vector space to reduce the learning difficulty imposed by the exponential dimension in the quantum Hilbert space. *PQK*s by definition, on the other hand, continue to be evaluated in the increasingly huge quantum Hilbert space. Below are some simple choices based on reduced density matrices (RDMs) of the quantum state.

- **The use of 1-RDMs to create a linear kernel function.**

$$Q_l^1(x_i,x_j) = \sum_k Tr[Tr_{m\neq k}[\rho(x_i)]Tr_{n\neq k}[\rho(x_j)]] \tag{7.13}$$

  where $Tr_m \neq k(\rho)$ represents a partial trace of the quantum state ρ over all qbits except the $k-th$ qubit. Any observable that can be expressed as a sum of one-body terms could be learned by it.

- **1-RDMs are used to create a Gaussian kernel function.**

$$Q_g^1(x_i,x_j) = exp(-\gamma\sum_k(Tr_{m\neq k}[\rho(x_i)] - Tr_{n\neq k}[\rho(x_j)])^2) \tag{7.14}$$

  where $\gamma > 0$ represents here as a hyper-parameter. Any nonlinear function of the 1-RDMs could be learned by it.

- **kRDMs are used to create a linear kernel.** $k-RDMs$ are used to create linear kernel. Mathematically, it is expressed as.

$$Q_l^k(x_i,x_j) = \sum_{K\in S_k(n)} Tr[Tr_{n\notin k}[\rho(x_i)]Tr_{m\notin k}[\rho(x_j)]] \tag{7.15}$$

  where $S_k(n)$ represents the set of subsets of k qbits from n, and $Tr_{n\notin K}$ denotes a partial trace of qbits not in subset K. Any observable that can be expressed as a sum of k-body terms could be learned.

The basic approach uses three steps to generate quantum features. The steps includes.

1. Datasets are preprocessed in this step to create a data set with fewer dimensions. A total of 5 snapshots were analyzed.

2. To relabel the dataset, embed this preprocessed data in quantum circuits. The Projected Quantum Kernel (PQK) feature can be computed after the dataset has been relabeled.

3. A standard machine learning model is trained after obtaining the *PQK* enhanced feature set, also known as quantum enhanced machine learning-based prediction model, on the re-labeled data sets. Training and testing sets are separated after projection.

The individual steps of this workflow are defined as follows:

- **Data preprocessing.** The dimensionality of datasets are lowered during the data preprocessing procedure. Data sets are preprocessed in this step to create a data set with fewer dimensions. The feature was reduced using Principal Component Analysis (PCA). The number of features in PCA is reduced to ten (20 originally due to five snapshots and five individual link prediction features, CN, JC, PA, AA, and SP). After PCA, each feature is transformed into 3d qubits. One more qubit is added, bringing the total number of qubits to 11. PQK features = $11*3$(3 for the qubit dimension= 33). The data preprocessing process includes obtaining the dataset, importing the appropriate libraries, importing the dataset, identifying and handling missing values, splitting the dataset, and feature scaling.

- **Computation of PQK features and Relabeling.** We will now build “stilted” quantum datasets by adding quantum components and re-labeling in a more condensed form. We’ll first produce PQK features and then relabel outputs based on their values to obtain the largest separation between quantum and classical procedures.

1. **PQK features and quantum encoding.** Based on $x-train$, $y-train$, $x-test$, and $y-test$, We're going to make a different set of features. On all qubits, It is defined as the 1-RDM on all qubits of the following.

$$V(x_{train}/n_{trotter})^{n_{trotter}}U_{1qb}|0\rangle \tag{7.16}$$

where U_{1qb} is a single qubit rotation's wall and

$$V(\hat{\theta}) = e^{-i\sum_i \hat{\theta}_i(X_iX_{i+1}+Y_iY_{i+1}+Z_iZ_{i+1})} \tag{7.17}$$

First, we generate the wall of single qubit rotations by using a for loop for qubits and their rotations. Then, we prepare $V(\hat{\theta})$ utilizing $tfq \cdot util \cdot exponential$ which is capable of exponentiating any commuting $cirq \cdot PauliSum$ objects. we use pauli X, Y and Z gates.

We now have all of the components necessary to construct complete encoding circuits. In this scenario, The number of qubits is kept at the same level as the number of features, which is 10. We create the above-mentioned random single qubit rotation wall with size (number of qubits, 3) and parameterized V using $V(\hat{\theta})$. The data is then converted into tensors using the resolve parameters of the tensor flow quantum library. Then, using the 1-RDM of the dataset shown above, we compute the *PQK* features and save the results in RDM, a tensor with dimension 33 (number of samples, number of qubits, 3).

$$rdm[i][j][k] = \langle \psi_i OP_j{}^K \psi_i \rangle \tag{7.18}$$

where i represents data points indexes, j represents qubits indexes and k is indexed on $\{\hat{X}, \hat{Y}, \hat{Z}\}$.

2. **PQK features relabeling.**

> We have the quantum generated features in $x - train - pqk$ and $x - test - pqk$. Based on the spectrum information in the dataset, we can label it. $x - train - pqk$ and $x - test - pqk$.

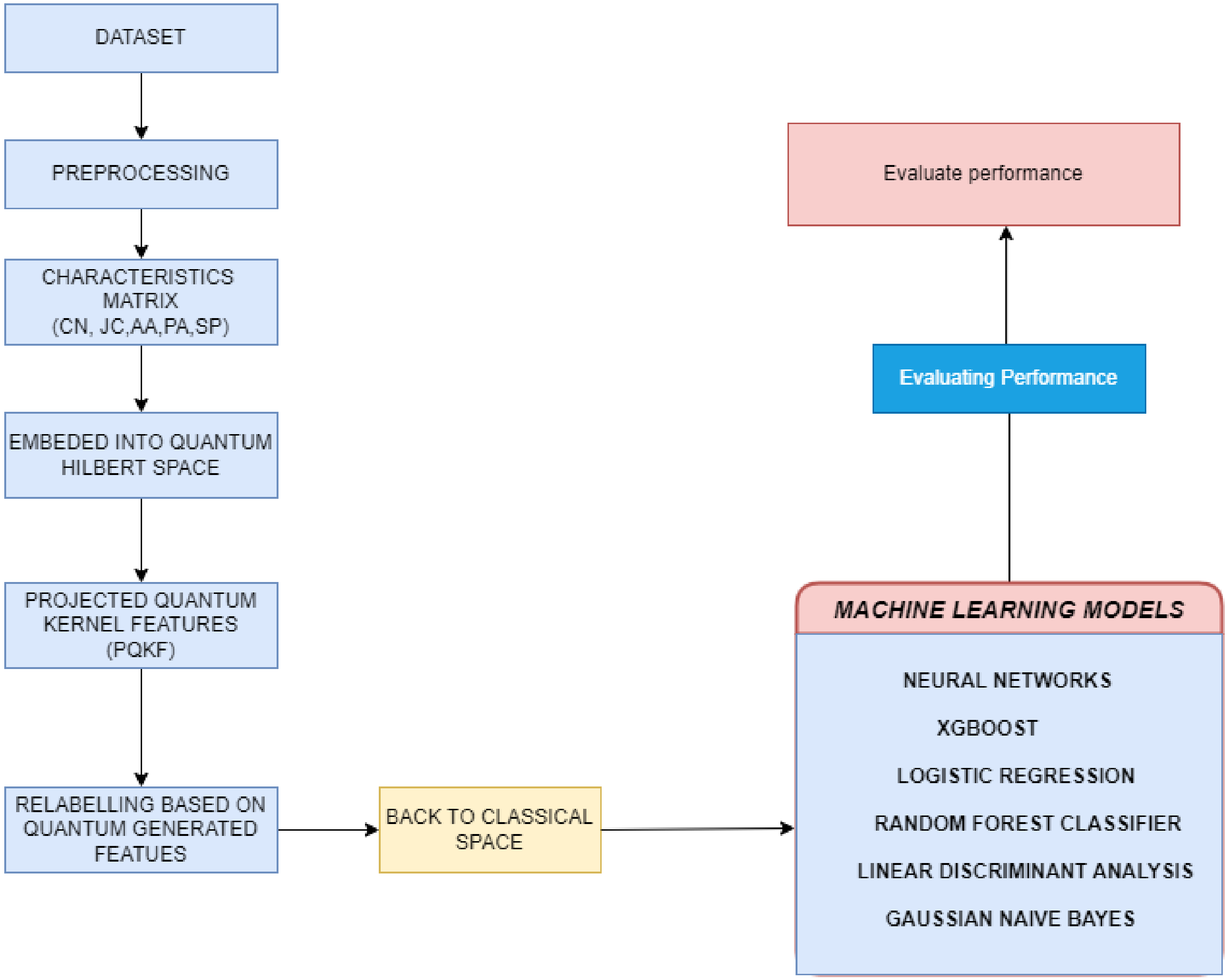
7.2 Proposed Method

Recent research has focused mostly on network topology to infer feature sets. These characteristics are non-domain-specific and general, therefore they may be implemented in any network [131, 150, 151]. Another research focuses on finding key node and edge information for improving the effectiveness of link prediction. These traits include topological, neighbourhood, and path-based characteristics [152, 153]. Several related studies [154, 207] indicate that the clustering coefficient is also closely associated with the link prediction issue. The complexity of link prediction is characterised as a binary classification issue. The categorization of a class is determined by the existence or absence of links. The label is assigned value 1 if there is a link between two nodes; conversely, it is set to 0.

7.2.1 Algorithms used in the proposed framework

This section will go over the numerous algorithms used in this work. The methods employed in this paper are feature generation techniques for classical machine learning models in the proposed model and quantum feature enhancement algorithms. We explained the feature generation of the suggested model in the feature generation algorithm for classical machine learning models. We used a variety of algorithms in the quantum feature generation process [208], including an algorithm for producing a wall of single qubit rotations, the creation of a circuit that produces $V(\hat{\theta})$, kernel matrix computation, for the production of PQK feature circuits around an input dataset, to get

FIGURE 7.3: Projected Quantum Kernel based Link Prediction Model Overview



Algorithm 6: Feature generation algorithm for proposed *PQKLP* approach

```
Input: Dynamic network $D$, number of snapshots $m$
Output: Feature Set ($edge\_fs$)
$all\_edges \leftarrow \{\}$                                  ▷ Initialization Phase
$true\_edges \leftarrow \{\}$
$false\_edges \leftarrow \{\}$
$true\_edges \leftarrow true\_edges(G_m)$    ▷ Edge dictionary for combining feature set between snapshots
$false\_edges \leftarrow randomized\_false\_edges(G_m)$
$all\_edges \leftarrow true\_edges + false\_edges$
$edge\_fs \leftarrow dict$
for $t \leftarrow 0$ to $m-1$ do
    $G \leftarrow t\_graph[t]$
    for $edge$ in $all\_edges$ do
        $FeatureSet \leftarrow \{\}$
        $i \leftarrow edge[0], j \leftarrow edge[1]$
        $FeatureSet \leftarrow append\ CommonNeighbour(G,i,j)$
        $FeatureSet \leftarrow append\ JaccardCoefficient(G,i,j)$
        $FeatureSet \leftarrow append\ Adamic/Adar(G,i,j)$
        $FeatureSet \leftarrow append\ PrefentialAttachment(G,i,j)$
        $FeatureSet \leftarrow append\ ShortestPath(G,i,j)$
        if $edge\_fs(edge)$ not empty then
            $temp \leftarrow edge\_fs[edge]$                   ▷ For last snapshot
        else
            $temp \leftarrow \{\}$
        $edge\_fs[edge] \leftarrow temp + FeatureSet$
$edge\_fs\_reduce \leftarrow PCA(edge\_fs)$   ▷ Transforming topological based feature set into informationally denser
$edge\_fs\_pqk \leftarrow PQK(edge\_fs\_reduce)$    ▷ Transforming reduced feature set into its quantum representation
return $edge\_fs\_pqk$                                   ▷ Return feature set
**This enhanced feature set is used with classical machine learning models for training and testing
```

PQK features, to generate kernel data point eigenvalues and eigenvectors, and to generate new labels that maximize geometric distance between kernels.

7.2.2 Feature generation algorithm for proposed PQKLP approach

In the proposed framework, Algorithm 6 illustrates the steps involved in feature generation for a traditional machine learning model and then its transformation using PQK. Line 1-3, is the initialization phase. all_edges, $true\_edges$, $false\_edges$ and $true\_edges$ are initialized as empty lists. $true\_edges$ and $randomized_{f}alse_{e}dges$ of the snapshots are stored in $true\_edges$ and $false\_edges$ in Line 4 and 5. In Line 6, $true\_edges$ and $false\_edges$ are stored in combination as all_edges. An empty dictionary $edge\_fs$ is created in Line 7 which with iteratively store edge features calculated on individual snapshots. Line 8, the for loop iterates to $m-1$ snapshots. Line 10-22, is used to generate features using various similarities indices like CN, JC, PA, AA and SP. It will loop over all_edges for each snapshot in line 10. It will append similarity measures derived by similarity techniques listed above. After calculating topological feature set over all snapshots, PCA transformation is applied to the feature set to generate an informationaly dense representation, $edge\_fs\_reduce$ (line 23), which is then projected into quantum vector space using PQK in line 24 ($edge\_fs\_pqk$). Finally, in line 25, $edge\_fs\_pqk$, feature set is returned. This enhanced feature set is used with classical machine learning models for training and testing to produce performance evaluation of the proposed PQKLP technique.

Computational Complexity

Let N be the size of the dataset, m be the number of snapshots, then the number of real edges in the last snapshot would be N/m, remember that we add a similar amount of randomized $false\_edges$ to the final classification edges i.e. all_edges, that makes it $2*(N/m)$. During the feature extraction phase, we first iterate over initial $m-1$ snapshots (leaving the final snapshot for classification), each time traversing over the $2*(N/m)$

classification edges and for each of the edges, we try to populate the features. The network metrics used as features and their complexities are Common Neighbor $O(k^2)$, Jaccard Coefficient $O(k^3)$, Adamic Adar $O(k^3)$, Preferential Attachment $O(k^2)$ and Shortest Path $O(ElogV)$ Considering V: number of nodes, k: degree of the nodes, E: number of edges. Note that the complexity is only for two fixed pairs of nodes not for all pairs hence an iteration of size V can be removed. It is clear that k $\ll$ E, hence taking the maximum time complexity for the shortest path i.e. $O(V\log V)$. Now, overall complexity of the algorithm comes out to be of order

$$O(m*2(N/m)*(E\log V)) = O(2N*E\log V) \approx O(N*E\log V). \tag{7.19}$$

7.3 Result Analysis

The results will be discussed and analyzed in this section. First, we compare the performance of our proposed PQKLP model with its component individual link prediction algorithms using six different machine learning models. All these methods use quantum kernel-based feature transformation. Finally, we compare and contrast the performance of the proposed PQKLP model with that of state-of-the-art approaches.

7.3.1 Performance comparison and analysis of PQKLP model with its component individual link prediction methods using Neural Network (NN)

Table 7.1 compares the performance of our proposed PQKLP model with its component individual link prediction methods using Neural Network (NN). In the Accuracy metric, our proposed model PQKLP is the best performing one in all datasets except Mit. In the Mit dataset, PQKLP is marginally outperformed by CN, JC, and AA. In the AUC metric,

TABLE 7.1: Performance comparison and analysis of PQKLP model with its component individual link prediction methods using Neural Network (NN)

	Data set	SP	CN	JC	AA	PA	PQKLP
Accuracy	Mit	0.5282	0.8094	0.7910	0.8377	0.7563	0.7746
	Fb-forum	0.6303	0.6397	0.6669	0.6566	0.6586	0.9124
	Radoslaw-email	0.5360	0.8320	0.8415	0.8270	0.7365	0.8755
	Eu-core	0.5400	0.8875	0.9130	0.9130	0.7025	0.9154
	CollegeMsg	0.5452	0.6112	0.5655	0.5980	0.6376	0.7289
AUC	Mit	0.5574	0.8938	0.8672	0.9151	0.7682	0.8662
	Fb-forum	0.7317	0.6346	0.6840	0.6891	0.6995	0.9437
	Radoslaw-email	0.5793	0.9050	0.9106	0.8987	0.7426	0.9293
	Eu-core	0.5998	0.9519	0.9687	0.9688	0.7288	0.9454
	CollegeMsg	0.6199	0.6108	0.5773	0.6151	0.6830	0.8511
F1 score	Mit	0.5091	0.8319	0.8153	0.8474	0.7587	0.7662
	Fb-forum	0.7096	0.5742	0.6108	0.6255	0.7222	0.9136
	Radoslaw-email	0.5931	0.8468	0.8575	0.8391	0.7678	0.8777
	Eu-core	0.5900	0.8956	0.9166	0.9122	0.7529	0.9151
	CollegeMsg	0.6732	0.4283	0.6106	0.4423	0.7157	0.7746
Precision	Mit	0.4528	0.8135	0.7750	0.8235	0.8630	0.7709
	Fb-forum	0.7414	0.7002	0.7511	0.6872	0.6651	0.9280
	Radoslaw-email	0.6566	0.8056	0.8246	0.8098	0.7589	0.8697
	Eu-core	0.6657	0.8704	0.9053	0.9125	0.6484	0.9063
	CollegeMsg	0.6593	0.7688	0.6950	0.7997	0.6095	0.8373

PQKLP is the best performing method in Fb-forum, Radoslaw-email, and CollegeMsg datasets. In the remaining Mit and Eu-core datasets, our method is marginally outperformed by CN, JC, and AA. In the F1 score metric, our method PQKLP is the best performing method in all datasets except Mit and Eu-core. In the Mit dataset, our method is marginally outperformed by CN, JC, and AA, while in the Eu-core dataset, it is only the second-best behind AA. In the Precision metric, our method PQKLP is the best performing method in all datasets except Mit and Eu-core. In the Mit dataset, our method is marginally outperformed by CN, JC, and AA, while in the Eu-core dataset, it is only the second-best behind AA. Overall we can conclude that our algorithm has good performance in all datasets except Mit when using NN-based classifier.

TABLE 7.2: Performance comparison and analysis of PQKLP model with its component individual link prediction methods using Logistic Regression (LR)

	Data set	SP	CN	JC	AA	PA	PQKLP
	Mit	0.5726	0.8321	0.7799	0.8520	0.8053	0.6707
	Fb-forum	0.7103	0.6345	0.6069	0.6310	0.7927	0.8800
Accuracy	Radoslaw-email	0.5410	0.8250	0.8205	0.8185	0.7600	0.7985
	Eu-core	0.6015	0.9070	0.8860	0.9150	0.7900	0.8823
	CollegeMsg	0.6294	0.6081	0.5594	0.6335	0.6873	0.7279
	Mit	0.5527	0.9037	0.8606	0.9262	0.8571	0.7162
AUC	Fb-forum	0.7390	0.6685	0.6585	0.6679	0.8987	0.9339
	Radoslaw-email	0.5590	0.9009	0.8909	0.8998	0.8702	0.8674
	Eu-core	0.6117	0.9720	0.9643	0.9723	0.9096	0.9280
	CollegeMsg	0.6128	0.6300	0.5980	0.6416	0.7210	0.7585
	Mit	0.6770	0.8296	0.7763	0.8481	0.8050	0.6700
F1 score	Fb-forum	0.6429	0.5754	0.5177	0.5291	0.7526	0.8832
	Radoslaw-email	0.6509	0.8261	0.8295	0.8253	0.7542	0.7992
	Eu-core	0.4417	0.9025	0.8798	0.9126	0.7578	0.8805
	CollegeMsg	0.3973	0.4521	0.3757	0.4789	0.5995	0.7720
	Mit	0.5434	0.8692	0.7985	0.8879	0.8570	0.6350
Precision	Fb-forum	0.8746	0.7067	0.6768	0.7024	0.9119	0.8866
	Radoslaw-email	0.5275	0.8516	0.8501	0.8533	0.8124	0.7992
	Eu-core	0.7221	0.9482	0.9513	0.9587	0.8845	0.8746
	CollegeMsg	0.9100	0.8636	0.7796	0.8507	0.8513	0.8373

7.3.2 Performance comparison and analysis of PQKLP model with its component individual link prediction methods using Logistic Regression (LR)

Table 7.2 compares the performance of our proposed PQKLP method with its component individual link prediction methods using Logistic Regression (LR). For the Accuracy metric, PQKLP is the best performing method in Fb-forum and CollegeMsg datasets. For Radoslaw-email and Eu-core datasets, our method outperforms the CN, JC, and AA algorithms. For the remaining Mit dataset, our method is outperformed by CN, JC, AA, and PA algorithms. For the AUC metric, our proposed PQKLP is the best performing algorithm in Fb-forum and CollegeMsg datasets. In Mit and Radoowslaw-email datasets, our algorithm is outperformed by CN, JC, AA, and PA algorithms, while in the remaining Eu-core dataset, it outperforms only SP and PA algorithms. In the F1 score metric, PQKLP is the best performing algorithm in Fb-forum and CollegeMsg datasets.

TABLE 7.3: Performance comparison and analysis of *PQKLP* model with its component individual link prediction methods using Linear Discriminant Analysis (*LDA*)

	Data set	SP	CN	JC	AA	PA	PQKLP
Accuracy	Mit	0.5706	0.8156	0.7727	0.8148	0.8604	0.6764
	Fb-forum	0.6961	0.6066	0.5970	0.6152	0.7397	0.8710
	Radoslaw-email	0.5510	0.8215	0.8280	0.8030	0.7745	0.8000
	Eu-core	0.5900	0.8610	0.8545	0.8725	0.7600	0.8750
	CollegeMsg	0.6081	0.6020	0.5695	0.5959	0.7096	0.7239
AUC	Mit	0.5273	0.9124	0.8701	0.9015	0.9292	0.7295
	Fb-forum	0.7299	0.6399	0.6270	0.6747	0.8687	0.9314
	Radoslaw-email	0.5646	0.9137	0.9037	0.8964	0.8845	0.8678
	Eu-core	0.6111	0.9596	0.9554	0.9713	0.9089	0.9242
	CollegeMsg	0.5933	0.6136	0.6019	0.5978	0.7582	0.7306
F1 score	Mit	0.6912	0.8033	0.7750	0.8116	0.8516	0.6752
	Fb-forum	0.6008	0.5333	0.5231	0.5147	0.6935	0.8738
	Radoslaw-email	0.6519	0.8198	0.8360	0.8020	0.7701	0.8008
	Eu-core	0.4415	0.8413	0.8417	0.8576	0.7105	0.8723
	CollegeMsg	0.3616	0.4574	0.3539	0.4125	0.5909	0.7699
Precision	Mit	0.5593	0.8583	0.8039	0.8412	0.9116	0.6416
	Fb-forum	0.8536	0.7064	0.6731	0.7246	0.8729	0.8855
	Radoslaw-email	0.5465	0.8657	0.8490	0.8463	0.8308	0.8011
	Eu-core	0.7505	0.9583	0.9499	0.9725	0.9078	0.8700
	CollegeMsg	0.8521	0.8238	0.8073	0.7878	0.8799	0.8339

In Radoslaw-email and Eu-core datasets, our algorithm outperforms the CN, JC, and AA algorithms. In the Mit dataset, the performance of our algorithm is only better than the SP algorithm. In the Precision metric, in Mit, Radoslaw-email, and Eu-core datasets, the performance of our algorithm is only better than the SP algorithm. In the remaining Fb-forum and CollegeMsg datasets, our algorithm is only slightly outperformed by the algorithms with the best performance. Overall we can conclude that the performance of our algorithm is suitable only for Fb-forum and CollegeMsg datasets while using a LR-based classifier.

7.3.3 Performance comparison and analysis of PQKLP model with its component individual link prediction methods using Linear Discriminant Analysis (LDA)

Table 7.3 compares the performance of our proposed PQKLP method with its component individual link prediction methods using Linear Discriminant Analysis (LDA). In the Accuracy metric, our proposed method performs best in Fb-forum, Eu-core, and CollegeMsg datasets. In Radoslaw-email, it is outperformed by CN, JC, and AA algorithms, while in the Mit dataset, it only outperforms the SP algorithm. In the AUC metric, PQKLP is the best performing algorithm in the Fb-forum dataset and the second-best performing one in the CollegeMsg dataset, just behind the PA algorithm. In Mit and Radoslaw-email datasets, our algorithm only outperforms the SP algorithm, while in the Eu-core dataset, it is the fourth-best performing algorithm. For the F1 score metric, our algorithm performs best in Fb-forum, Eu-core, and CollegeMsg datasets. In the Mit dataset, it only outperforms the SP algorithm, while in the Radoslaw-email algorithm, it outperforms both SP and PA algorithms. For the Precision metric, our proposed PQKLP is the best performing algorithm in the Fb-forum dataset and the second-best performing algorithm in the CollegeMsg dataset, just behind the PA algorithm. For the remaining datasets, our algorithm only outperforms the SP algorithm. Overall we can conclude that the performance of our algorithm is suitable only for Fb-forum and CollegeMsg datasets while using a LDA-based classifier.

7.3.4 Performance comparison and analysis of PQKLP model with its component individual link prediction methods using XGBoost (XGB)

Table 7.4 compares the performance of our proposed PQKLP method with its component individual link prediction methods using XGBoost (XGB). In the Mit dataset for all four

TABLE 7.4: Performance comparison and analysis of PQKLP model with its component individual link prediction methods using XGBoost (XGB)

	Data set	SP	CN	JC	AA	PA	PQKLP
Accuracy	Mit	0.5717	0.8461	0.8072	0.8351	0.8517	0.7442
	Fb-forum	0.7345	0.6510	0.6483	0.6510	0.8152	0.9000
	Radoslaw-email	0.5640	0.8185	0.8375	0.8310	0.7940	0.8455
	Eu-core	0.6045	0.9120	0.9090	0.9075	0.8075	0.8892
	CollegeMsg	0.6132	0.6102	0.6020	0.6345	0.7695	0.7827
AUC	Mit	0.5661	0.8464	0.8088	0.8354	0.8530	0.7505
	Fb-forum	0.7341	0.6516	0.6451	0.6483	0.8152	0.9006
	Radoslaw-email	0.5482	0.8146	0.8368	0.8298	0.7916	0.8455
	Eu-core	0.5977	0.9123	0.9084	0.9080	0.8065	0.8894
	CollegeMsg	0.6230	0.6067	0.6067	0.6353	0.7647	0.8072
F1 score	Mit	0.6870	0.8538	0.8172	0.8463	0.8554	0.7543
	Fb-forum	0.6654	0.6174	0.5742	0.5923	0.8217	0.9015
	Radoslaw-email	0.6727	0.8360	0.8464	0.8442	0.8112	0.8470
	Eu-core	0.4021	0.9137	0.9111	0.9069	0.8143	0.8866
	CollegeMsg	0.4322	0.4228	0.4244	0.4698	0.7990	0.7988
Precision	Mit	0.5510	0.8454	0.7901	0.8416	0.8258	0.7125
	Fb-forum	0.8945	0.6907	0.7042	0.6949	0.8261	0.9142
	Radoslaw-email	0.5558	0.8123	0.8123	0.7942	0.7823	0.8445
	Eu-core	0.7768	0.9135	0.9063	0.8984	0.7987	0.8928
	CollegeMsg	0.8915	0.7973	0.8117	0.8576	0.7453	0.8839

performance evaluation metrics, our algorithm only outperforms the SP algorithm, while in the Eu-core dataset, our algorithm outperforms both SP and PA algorithms. In the three remaining datasets, Fb-forum, Radoslaw-email, and CollegeMsg, our algorithm PQKLP has the best performance across all evaluation metrics. Overall we can conclude that the performance of our algorithm is suitable for Fb-forum, Radoslaw-email, and CollegeMsg datasets while using a XGB-based classifier.

7.3.5 Performance comparison and analysis of PQKLP model with its component individual link prediction methods using Random Forest Classifier (RFC)

Table 7.5 compares the performance of our proposed PQKLP method with its component individual link prediction methods using Random Forest Classifier (RFC). For the Accuracy metric, our algorithm PQKLP performs best in Fb-forum and Radoslaw-email

TABLE 7.5: Performance comparison and analysis of PQKLP model with its component individual link prediction methods using Random Forest Classifier (RFC)

	Data set	SP	CN	JC	AA	PA	PQKLP
	Mit	0.5474	0.8173	0.8347	0.8373	0.8620	0.7906
	Fb-forum	0.7184	0.6531	0.6531	0.6310	0.8117	0.9041
Accuracy	Radoslaw-email	0.5700	0.8055	0.8285	0.8350	0.7945	0.8605
	Eu-core	0.6015	0.9050	0.9045	0.9110	0.8100	0.9062
	CollegeMsg	0.6193	0.6193	0.6264	0.6193	0.7838	0.7827
	Mit	0.5597	0.9123	0.9086	0.9148	0.9400	0.8622
AUC	Fb-forum	0.7387	0.6726	0.6597	0.6281	0.8907	0.9391
	Radoslaw-email	0.5829	0.8749	0.8996	0.9025	0.8683	0.9209
	Eu-core	0.6063	0.9514	0.9570	0.9603	0.8865	0.9411
	CollegeMsg	0.6393	0.6253	0.6179	0.6270	0.8794	0.8827
	Mit	0.6535	0.8179	0.8376	0.8383	0.8622	0.7809
F1 score	Fb-forum	0.6506	0.6086	0.5740	0.5312	0.8181	0.9069
	Radoslaw-email	0.6722	0.8167	0.8437	0.8460	0.8111	0.8651
	Eu-core	0.4138	0.9046	0.9056	0.9121	0.8179	0.9055
	CollegeMsg	0.4241	0.4330	0.4235	0.4476	0.8044	0.7994
	Mit	0.5215	0.8074	0.8072	0.8439	0.8362	0.7656
Precision	Fb-forum	0.8874	0.7001	0.7702	0.6967	0.7970	0.9132
	Radoslaw-email	0.5570	0.7990	0.8303	0.8193	0.7785	0.8601
	Eu-core	0.7762	0.9057	0.9017	0.9014	0.8093	0.9028
	CollegeMsg	0.9337	0.8180	0.7937	0.8530	0.7253	0.8842

datasets. In CollegeMsg, it is just slightly behind the PA algorithm, while in the Eur-core dataset, it is only slightly behind the AA algorithm. In the Mit dataset, our algorithm only outperforms the SP algorithm. For the AUC metric, our proposed PQKLP performs best in Fb-forum, Radoslaw-email, and CollegeMsg datasets. In the Mit dataset, it only outperforms the SP algorithm, while in the Eu-core dataset, it outperforms both SP and PA algorithms. For the F1 score metric, PQKLP performs best in Fb-forum and Radoslaw-email datasets. In the CollegeMsg dataset, it is only slightly outperformed by the PA algorithm. In the Mit dataset, it only outperforms the SP algorithm, while in the Eu-core dataset, it outperforms both SP and PA algorithms. For the Precision metric, our algorithm performs best in Fb-forum, Radoslaw-email, and CollegeMsg datasets. In the Eu-core dataset, it is only slightly outperformed by the CN algorithm, but in the Mit dataset, it only outperforms the SP algorithm. Overall we can conclude that the performance of our algorithm is suitable for Fb-forum, Radoslaw-email, Eu-core, and CollegeMsg datasets while using a RFC-based classifier.

TABLE 7.6: Performance comparison and analysis of PQKLP model with its component individual link prediction methods using Gaussian Naive Bayes (GNB)

	Data set	SP	CN	JC	AA	PA	PQKLP
Accuracy	Mit	0.5614	0.8323	0.7744	0.8020	0.8460	0.6851
	Fb-forum	0.6894	0.6315	0.6321	0.6041	0.7324	0.8876
	Radoslaw-email	0.5570	0.7975	0.8090	0.8220	0.7555	0.8290
	Eu-core	0.6160	0.8920	0.8610	0.8925	0.7235	0.8818
	CollegeMsg	0.5898	0.6162	0.5299	0.6396	0.6670	0.7269
AUC	Mit	0.5669	0.8988	0.8638	0.8826	0.9147	0.7702
	Fb-forum	0.7245	0.6778	0.6724	0.6752	0.8779	0.9391
	Radoslaw-email	0.5398	0.8913	0.8817	0.9147	0.8836	0.8947
	Eu-core	0.6287	0.9684	0.9609	0.9663	0.8432	0.9316
	CollegeMsg	0.6051	0.6099	0.5263	0.6398	0.8138	0.7942
F1 score	Mit	0.6791	0.8314	0.7783	0.7961	0.8359	0.6964
	Fb-forum	0.5919	0.4615	0.5037	0.4227	0.6659	0.8907
	Radoslaw-email	0.6639	0.7922	0.8128	0.8127	0.7456	0.8291
	Eu-core	0.4466	0.8815	0.8507	0.8825	0.6402	0.8790
	CollegeMsg	0.3892	0.3908	0.4107	0.4459	0.5591	0.7727
Precision	Mit	0.5474	0.8414	0.7678	0.8234	0.8818	0.6483
	Fb-forum	0.8969	0.8304	0.7275	0.8018	0.9018	0.8916
	Radoslaw-email	0.5588	0.8444	0.8200	0.8788	0.8472	0.8365
	Eu-core	0.8010	0.9410	0.9597	0.9484	0.8880	0.8843
	CollegeMsg	0.8498	0.8296	0.7763	0.8616	0.9170	0.8332

7.3.6 Performance comparison and analysis of PQKLP model with its component individual link prediction methods using Gaussian Naive Baiyes classifier (GNB)

Table 7.6 compares the performance of our proposed PQKLP method with its component individual link prediction methods using the Gaussian Naive Bayes classifier (GNB). For the Accuracy metric, our proposed algorithm PQKLP performs best in Fb-forum, Radoslaw-email, and CollegeMsg datasets. In the Eu-core dataset, PQKLP has the third-best performance, and it only outperforms SP in the Mit dataset. For the AUC metric, PQKLP has the best performance in the Fb-forum dataset and second-best performance in Radoslaw-email and CollegeMsg datasets. In the Mit dataset, PQKLP only outperforms the SP algorithm, while in the Eu-core dataset, it outperforms both SP and PA algorithms. For the F1 score metric, our proposed PQKLP performs best in Fb-forum, Radoslaw-email, and CollegeMsg datasets. In the Eu-core dataset, it is only

TABLE 7.7: Performance Comparison of link prediction in different machine learning models with proposed projected Quantum machine learning model

	Datasets	NN	Q-NN	XGB	Q-XGB	LR	Q-LR	RFC	Q-RFC	LDA	Q-LDA	GNB	Q-GNB
Accuracy	Mit	0.7608	**0.7746**	**0.8797**	0.7442	**0.8116**	0.6707	**0.8986**	0.7906	**0.8623**	0.6764	**0.8333**	0.6851
	Fb-forum	0.7861	**0.9124**	0.8413	**0.9000**	0.7552	**0.8800**	0.8034	**0.9041**	0.8207	**0.8710**	0.8069	**0.8876**
	Radoslaw-email	0.7175	**0.8755**	**0.8495**	0.8455	0.7875	**0.7985**	0.8600	**0.8605**	**0.8350**	0.8000	0.8075	**0.8290**
	Eu-core	0.8220	**0.9154**	**0.9235**	0.8892	0.8425	**0.8823**	**0.9500**	0.9062	**0.9175**	0.8750	**0.9225**	0.8818
	CollegeMsg	**0.7310**	0.7289	**0.7929**	**0.7827**	0.6650	0.7279	**0.8274**	**0.7827**	0.6751	**0.7239**	0.6751	**0.7269**
AUC	Mit	0.7671	**0.8662**	**0.8800**	0.7505	**0.9005**	0.7162	**0.9345**	0.8622	**0.9096**	0.7295	**0.8638**	0.7702
	Fb-forum	0.7885	**0.9437**	0.8412	**0.9006**	0.8624	**0.9339**	0.8864	**0.9391**	0.8692	**0.9314**	0.8825	**0.9391**
	Radoslaw-email	0.7192	**0.9293**	**0.8486**	0.8455	**0.8758**	0.8674	**0.9355**	0.9209	**0.9222**	0.8678	**0.9123**	0.8947
	Eu-core	0.8326	**0.9454**	**0.9236**	0.8894	**0.9311**	0.9280	**0.9808**	0.9411	**0.9818**	0.9242	**0.9789**	0.9316
	CollegeMsg	0.8037	**0.8511**	0.7914	**0.8072**	0.7149	**0.7585**	0.8796	**0.8827**	**0.7318**	0.7306	0.7113	**0.7942**
F1 Score	Mit	**0.7967**	0.7662	**0.8823**	0.7543	**0.8143**	0.6700	**0.9054**	0.7809	**0.8652**	0.6752	**0.8435**	0.6964
	Fb-forum	0.7726	**0.9136**	0.8319	**0.9015**	0.7171	**0.8832**	0.8055	**0.9069**	0.7953	**0.8738**	0.7879	**0.8907**
	Radoslaw-email	0.7341	**0.8777**	**0.8594**	0.8470	0.7658	**0.7992**	0.8600	**0.8651**	**0.8308**	0.8008	0.7843	**0.8291**
	Eu-core	0.8311	**0.9151**	**0.9245**	0.8866	0.8174	**0.8805**	**0.9468**	0.9055	**0.9065**	0.8723	**0.9122**	0.8790
	CollegeMsg	**0.7768**	0.7746	**0.8120**	0.7988	0.5926	0.7720	**0.8482**	**0.7994**	0.6000	**0.7699**	0.5844	**0.7727**
Precision	Mit	0.7625	**0.7709**	**0.8699**	0.7125	**0.8906**	0.6350	**0.9306**	0.7656	**0.9385**	0.6416	**0.8732**	0.6483
	Fb-forum	0.8440	**0.9280**	0.8413	**0.9142**	0.8411	**0.8866**	0.7919	**0.9132**	**0.9182**	0.8855	0.8667	**0.8916**
	Radoslaw-email	0.7412	**0.8697**	0.8298	**0.8445**	**0.8323**	0.7992	0.8431	**0.8601**	**0.8351**	0.8011	**0.8696**	0.8365
	Eu-core	0.8296	**0.9063**	**0.9166**	0.8928	**0.8813**	0.8746	**0.9319**	0.9028	**0.9524**	0.8700	**0.9583**	0.8843
	CollegeMsg	0.6768	**0.8373**	0.7451	**0.8839**	**0.8571**	0.8373	0.8051	**0.8842**	**0.8889**	0.8339	**0.9375**	0.8332

marginally outperformed by CN and AA algorithms, while in the Mit dataset, it only outperforms the SP algorithm. Our algorithm has the worst relative performance for the Precision metric out of all performance evaluation metrics. PQKLP is the second-best performing algorithm in the Fb-forum dataset, third-best in the CollegeMsg dataset, and fourth-best in the Radoslaw-email dataset. For Mit and Eu-core datasets, our algorithm only outperforms the SP algorithm.

7.3.7 Performance comparison and analysis of classical machine learning model with proposed PQKLP-based machine learning model with same feature set

Table 7.7 compares the performance of classical and quantum machine learning models for the same feature set (CN,JC,PA,AA,&SP). We analysed the results of six classical machine learning models: NN, XGB, LR, RFC, LDA, and GNB. For better understanding, PKQLP is abbreviated as Q in this section. Q-NN outperforms NN in terms of Accuracy on the Mit dataset. Other than the Q-NN based model, all other quantum machine learning models perform worse than their non-quantum-based counterparts on the Mit dataset. In the Fb-forum dataset, all quantum-machine learning-based models perform much better than their non-quantum learning-based counterparts. The Q-NN algorithm outperforms NN on the Radoslaw-email dataset. The outcome of Q-XGB is comparable to that of XGB. The results of the Q-LR and LR pair and Q-RFC and RFC pair are nearly identical. Compared to Q-LDA, LDA performs better. In comparison to GNB, Q-GNB produces superior results. On the Eu-core dataset, Q-NN and Q-LR give better results than NN and LR, respectively. XGB, RFC, LDA, and GNB give better performance than Q-XGB, Q-RFC, Q-LDA, and Q-GNB respectively. On the CollegeMsg dataset, Q-NN and Q-XGB have marginally better performance than NN and XGB. Q-LR performs better than LR, and Q-RFC performs better than RFC. Compared to LDA and GNB, Q-LDA and Q-GNB produce better results.

In terms of AUC, Q-NN perform better than NN while all other quantum-based algorithms perform worse than their non-quantum-based counterparts on the Mit dataset. On the Radoslaw-email dataset, all quantum-based machine learning models perform better than their non-quantum-based counterparts. Even though Q-NN outperforms NN in the Eu-core dataset, XGB and LR pair have comparable performance to Q-XGB and Q-LR pair, respectively. RFC, LDA, and GNB have better performance than Q-RFC, Q-LDA, and Q-GNB. On the CollegeMsg dataset, all quantum-based models have better performance thair their non-quantum-based counterparts except LDA, which gives better performance than Q-LDA.

Regarding the F1 score, all non-quantum-based machine learning models give better results than their quantum-based machine learning counterparts on the Mit dataset. On the Fb-forum dataset, the situation becomes precisely the reverse of the Mit dataset, such that quantum-based machine learning models perform better. On the Radoslaw-email dataset, Q-NN, Q-LR, Q-RFC, and Q-GNB gives better performance than NN, LR, RFC, and GNB while other give worse. On the Eu-core dataset, Q-NN and Q-LR give better performance than NN and LR. Other non-quantum-based machine learning models perform better than their quantum-based counterparts. On the CollegeMsg dataset, Q-LR, Q-LDA, and Q-GNB have better performance than LR, LDA, and GNB. XGB and RFC have better performance than Q-XGB and Q-RFC while the remaining NN based pair is comparable.

In terms of Precision, Q-NN gives a superior result than NN on the Mit dataset. Other than this exception, all other non-quantum-based machine learning models perform better than quantum-based models. On the Fb-forum dataset, quantum-based machine learning models perform better than their non-quantum-based counterparts except for the LDA and Q-LDA pair, where this pattern is reversed. On the Radoslaw-email dataset, Q-NN, Q-XGB, and Q-RFC have superior results than NN, XGB, and RFC. LR, LDA, and GNB give better results than Q-LR, Q-LDA, and Q-GNB. On the Eu-core dataset, NN outperforms Q-NN, while the non-quantum-based machine learning models outperform the quantum-based ones for all other cases. On CollegeMsg dataset, Q-NN, Q-XGB, Q-RFC give better results than NN, XGB and RFC, and LR, LDA and GNB have better performance than Q-LR, Q-LDA and Q-XGB.

7.3.8 Performance comparison and analysis of Projected Quantum Kernel based Link Prediction (PQKLP) model with various state-of-the art algorithms

Table 7.8 compares the performance of the proposed PQKLP approach with four state-of-the-art algorithms. For the Accuracy metric, the results of our three best algorithm variations, i.e., PQKLP-RFC, PQKLP-XGB, and PQKLP-NN, are better than N2V, WEAK, and CTDNE algorithms, falling only slightly behind XGB. For the AUC metric, the results of the proposed PQKLP-RFC and PQKLP-NN are better than all state-of-the-art algorithms for all datasets except Eu-core. In the Eu-core dataset, N2V and WEAK slightly outperform the proposed algorithms. For the F1 score, the results of PQKLP-RFC, PQKLP-NN, and PQKLP-XGB are better than all the state-of-the-art

TABLE 7.8: Performance comparison and analysis of Projected Quantum Kernel based Link Prediction (PQKLP) model with various state-of-the art algorithms

	Datasets	N2V	WEAK	CTDNE	LGQ	PQKLP-RFC	PQKLP-XGB	PQKLP-NN
AUC	Mit	0.69057	0.75855	0.61008	0.83272	0.8622	0.7505	**0.8662**
	Fb-forum	0.85458	0.9182	0.70761	0.88449	**0.9391**	0.9006	**0.9437**
	Radoslaw-email	0.77295	0.90627	0.79913	0.81328	**0.9209**	0.8455	**0.9293**
	Eu-core	0.94527	0.97181	0.72765	0.91646	**0.9411**	0.8894	**0.9454**
	CollegeMsg	0.70554	0.67802	0.55264	0.63538	**0.8827**	0.8072	0.8511
F1 Score	Mit	0.43384	0.48768	0.39541	0.728	**0.7809**	0.7543	0.7662
	Fb-forum	0.55886	0.73225	0.39994	0.82953	0.9069	0.9015	**0.9136**
	Radoslaw-email	0.45745	0.66927	0.50486	0.72753	0.8651	0.8470	**0.8777**
	Eu-core	0.70746	0.82317	0.38405	0.84601	0.9055	0.8866	**0.9151**
	CollegeMsg	0.40279	0.35695	0.24717	0.41832	**0.7994**	**0.7988**	0.7746
Precision	Mit	0.29159	0.51329	0.29692	0.70149	0.7656	0.7125	**0.7709**
	Fb-forum	0.44561	0.8954	0.29888	0.87108	0.9132	0.9142	**0.9280**
	Radoslaw-email	0.3184	0.73699	0.37426	0.79833	**0.8601**	0.8445	**0.8697**
	Eu-core	0.57848	0.82944	0.25906	0.82414	**0.9028**	0.8928	**0.9063**
	CollegeMsg	0.33454	0.78033	0.18709	0.78376	**0.8842**	**0.8839**	0.8373
Accuracy	Mit	0.50947	0.7818	0.58761	**0.8722**	0.7906	0.7442	0.7746
	Fb-forum	0.80559	0.89733	0.69534	**0.94461**	0.9041	0.9000	**0.9124**
	Radoslaw-email	0.61177	0.83337	0.69551	**0.8993**	0.8605	0.8455	**0.8755**
	Eu-core	0.87149	0.93163	0.59771	**0.9477**	0.9062	0.8892	**0.9154**
	CollegeMsg	0.73844	0.62566	0.63071	**0.86546**	0.7827	0.7827	0.7289

algorithms across all datasets. The precision score evaluation metric also follows the same pattern like the F1 score, whereby the results of our proposed approaches are much better than state-of-the-art algorithms in all cases. Out of all the proposed quantum-based machine learning approaches, PQKLP-NN provides better results than PQKLP-RFC in most cases. It's worth noting that we only use features generated by five typical link prediction methods in our approach, but state-of-the-art methods like LGQ use several more. The number of dimensions used in embedding-based techniques is also quite significant. As a result, we achieve better results in our suggested PQKLP with considerably less data.

7.4 Conclusion

As a result of recent technological developments, the world is moving towards quantum computing. In this paper, we propose a novel strategy for addressing the link prediction problem using Projected Quantum Kernel (PQK) enhanced machine learning models, which utilize both local and global information for feature generation. The goal of our research is to create a quantum-assisted feature-based new approach for link prediction that integrates Projected Quantum Kernel (PQK) with machine learning models to increase prediction performance. By using PQK, we enhanced our data using high-dimensional Hilbert spaces to achieve improved link prediction. Quantum models look at data in high-dimensional Hilbert spaces, to which in other cases we can only have access through inner products revealed by measurements because they have a mathematical structure that is similar to that of quantum mechanics. We compared the experimental results of the proposed approach i.e Quantum enhanced Neural Network (PQKLP-NN), Quantum enhanced XGBoost (PQKLP-XGB), Quantum enhanced Logistic Regression (PQKLP-LR), Quantum enhanced Random forest classifier (PQKLP-RFC), Quantum enhanced Linear discriminant Analysis (PQKLP-LDA) and Quantum enhanced Gaussian Naive Bayes classifier (PQKLP-GNB) to those of corresponding classical machine learning models. As validation of our feature set choice, we also compared the results of the full feature set of link prediction with its individual components, i.e., CN, JC, PA, AA and SP. The experimental results demonstrate that the new quantum-assisted feature-based technique outperforms the corresponding machine learning models for some cases, especially for PQKLP-NN and PQKLP-RFC. We used four performance matrices on five well-known dynamic datasets to compare our quantum-assisted methodology to individual link prediction approaches (in quantum-enhanced setting) as well as state-of-the-art methods, demonstrating that the suggested approach outperforms them in most cases. We can build on this work in the future by experimenting with expanded feature sets, possibly embedding-based, and feature ranking based on their performance in this framework. This work can also be expanded to the domain of other types of networks such as multiplex and attributed ones.