

PQCLP: Parameterized quantum circuit based link prediction in dynamic networks

Nisha Singh^{a,*}, Mukesh Kumar^b, Bhaskar Biswas^a

^a Department of Computer Science and Engineering, Indian Institute of Technology (BHU), Varanasi, India

^b School of Computer Engineering, Kalinga Institute of Industrial Technology (KIIT), Deemed to be University, Bhubaneswar, Odisha, India

ARTICLE INFO

Dataset link: [Dynamic Dataset \(Reference data\)](#)

Keywords:

Link prediction
Dynamic networks
Quantum computing
Quantum machine learning
Parameterized circuits

ABSTRACT

Link prediction has been challenging, especially when the network is dynamic and complex. The most effective classical method for performing this task involved using machine learning algorithms with features taken from topological network indices. Even while these traditional ML algorithms perform better, they still require a lot of processing resources as the size and number of features in the network increase. This is the ideal situation where quantum computing may fit, as it provides impressive predictions and speedup arising out of quantum phenomena like superposition, entanglement, parallelization, and high dimensional space. Additionally, relatively little research has been done to examine the full potential of quantum computation for link prediction. A few of the earlier attempts are limited to projecting the features to quantum space and then using quantum-projected kernels with classical ML techniques or using hybrid classifiers by incorporating quantum enhancement in traditional random walks. We propose Parameterized Quantum Circuit based Link Prediction (PQCLP) model where we have used quantum circuits not only for projecting the classical data but also for training and optimization in quantum space using Variational Circuits Aka Ansatz which is a parameterized circuit. Here we employ two quantum methods namely Variational Quantum Classifier (VQC) and Quantum Neural Network Classifier (QNN) having classical equivalence with Support Vector Machines and Neural Networks respectively. We present here a detailed comparison of these models with their classical counterparts, within different feature categories and test ratios, and finally with a few state-of-the-art methods using several performance evaluation metrics.

1. Introduction

Network-based graphs have grown in popularity as a result of their ability to represent a variety of disciplines, including communication networks, social networks, bio-metrics, drug discovery, criminal networks, e-commerce, and knowledge graphs. They also aid in the formulation of solutions for a number of tasks, including the recommendation of friends and products [1], routing improvement [2], group/community detection [3], protein-protein interaction (PPI) [4], a graph summarizing [5], and question-answering [6]. The essence of link prediction lies in the identification of the most promising missing links within a network. The complexity of this task amplifies when dealing with dynamic networks that continually evolve over time. In such dynamic networks, existing connections may disappear while new links emerge, adding a layer of intricacy to the link prediction process. Time-series modeling and prediction is an excellent technique for studying link prediction in dynamic networks it changes over time. Link prediction in this technique predicts the existence of links by

observing behavioral patterns of these links over several graphs formed by dividing the network into several timestamps [7]. Eventually, our task is to predict the link in the next timestamp based on observations from previous timestamps.

Features that encapsulate the network behavior range from topological similarity based, node specific semantic attribute based, probabilistic or statistical, stochastic block model (SBM) [8,9]. Most recent researches extract features from network topology as their characteristics are generic and domain independent [10]. They also perform well as topological features are network characteristics determined from network links that establish the overall network structure [11,12]. In recent research generally, semantic methods are used to attribute topological similarity methods, a similar approach by Hasan et al. [13]. In our work, we use topological similarity based indices to extract features that range from local, global to quasi-local so as to maintain the trade-off between performance and computation cost. In Section 3, which covers the preliminaries, the three similarity indices—local, global, and quasi-local—are explained in detail.

* Corresponding author.

E-mail addresses: nishasingh.rs.cse21@itbhu.ac.in (N. Singh), mukesh.kumarfcs@kiit.ac.in (M. Kumar), bhaskar.cse@itbhu.ac.in (B. Biswas).

We understand that probability theory serves as the fundamental framework for inference and statistical modeling in classical machine learning. Quantum mechanics, on the other hand, offers a broader perspective on probability theory, enabling efficient utilization in the realm of quantum machine learning [14]. This area has witnessed substantial research efforts, resulting in the emergence of various quantum machine learning approaches. One noteworthy development is Quantum Support Vector Machines (QSVM), introduced by Rebentrost et al. [15], designed to address least-squares SVM problems. Quantum Neural Networks (QNN) have also gained prominence with diverse architectural variations being explored across different domains [16–18]. Additionally, Cong et al. [19] introduced a novel Quantum Convolutional Neural Networks (QCNNs). In 2014, Wiebe et al. [20] introduced a quantum adaptation of the k-nearest neighbors algorithm. Furthermore, authors in [21] proposed Quantum Linear Regression as a quantum counterpart to classical linear regression. The quantum decision tree classifier, as devised by Lu et al. [22], additionally, a quantized variant of the k-means clustering method, known as Q-means, has been proposed with [23] showing comparable results and convergence characteristics to the traditional k-means algorithm.

Quantum computation's integration into link prediction tasks remains relatively unexplored. Some notable attempts include the work in [24], where the authors proposed the utilization of Hybrid Quantum–Classical Neural Networks. The objective here is to enhance embedding for more effective link prediction. In a different approach, Kumar et al. employed Projected Quantum Kernel to project features into quantum space and then back into classical space for use with classical machines [25]. Another area of investigation focuses on enhancing classical random walk methods with quantum enhancements [26,27]. However, it is important to note that none of these approaches fully leverage quantum computation or seamlessly utilize quantum capacities in all phases of learning.

In this article, we have made an attempt to fully utilize the quantum enhancements for machine learning. In our approach, we use these dynamic network characteristics and pass them to a quantum circuit to generate a Parameterized Quantum Circuit based Link Prediction (PQCLP) model. We employ quantum computation not only for feature projection into quantum space but also for conducting training and optimization tasks within the quantum realm with the use of Variational/Parameterized quantum circuits. With the projection of features, we get an advantage of (1) high dimensional complex vector space, the Hilbert Space [28]. The Hilbert space is equipped with an inner product that returns a scalar quantity representing the amount to which the first vector lies to the other vector [29] (2) entanglements to generate complex models that are hard to simulate classically [30] (3) parallelization arising from the parallel computation of each quantum superposition and faster quantum registers. Using quantum computing and present-day noisy intermediate-scale quantum technology [31], it is possible to speed up classical machine learning algorithms without compromising classification accuracy. Today's QML algorithms' fault tolerance is very low, still, it is worth exploring as the research on quantum hardware may result in new fault-tolerant devices. Listing out a few motivations and inspirations that lead us to work on this methodology:

- Earlier approaches of Quantum Link Prediction were restricted to use of only quantum projection of feature to complex vector space and extracting projected quantum kernels for use with conventional machine learning techniques [24,25,32] or using quantum enhancements in conventional link prediction techniques like random walk [26,27,33].
- Quantum machine learning is an upcoming and propitious research field that tends to combine the dramatic performance advantage offered by quantum paralleling and the ability of machine learning algorithms to learn complex distributions of high-dimensional data.

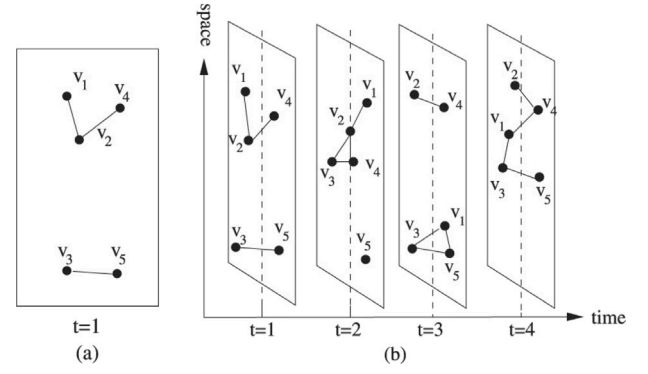


Fig. 1. Evolution of a dynamic network with time.

- Variational quantum algorithms (VQAs) are a prominent technique to leverage quantum computing for machine learning [34]. They train a parameterized quantum circuit using a classical optimizer. We have extended the concept and employed the VQA for link prediction with engineered network properties.
- There is a strong hope of achieving quantum supremacy in the NISQ (Near Term Intermediate Scale) era [35,36].
- We strongly believe that, by properly integrating quantum computation into the task of link prediction, we can obtain promising outcomes in terms of predictions and speedup thanks to the performance advantages demonstrated in prior research.

The major contributions of our work are:

- Our approach studies the impact of quantum optimization on the task of link prediction in dynamic networks. Unlike the previous methods, it uses quantum enhancements not only for feature set projection to quantum space but also for training and optimization in quantum space. Thus taking the added advantage naturally arising from quantum high dimensional Hilbert space and quantum paralleling in all the phases.
- The PQCLP model achieves quantum training with the help of Variational circuits/parameterized circuits aka Ansatz whose weights can be optimized with parameter shift techniques in quantum space.
- In PQCLP, we employ two QML models, Variational Quantum Circuit (VQC) and Quantum Neural Networks (QNN) having different learning approaches as they share structural characteristics with their classical equivalents, Support Vector Machines, and Neural Networks respectively.
- The PQCLP model takes into account a genuine set of topological features from all the categories considering the trade-off between accuracy and speed.
- The model presents a complete analysis with several performance metrics and a comparison between local, global, quasi-local, and combined predictions. Alongside a comparison of performance with different test-train ratios and with several state-of-art methods is also presented here.

2. Preliminaries

In dynamic networks, we consider the continuously changing dynamic structure of the network. Formally $G(V, E)$ is the graph of the dynamic network where V is the set of nodes and E the set of undirected edges, $\text{edge}(u, v)$ indicates the edge between u and v . Liben-Nowell et al. [37] assume that a graph $G_{t_0-t_1}(V, E)$ is a snapshot of a network at time $[t_0, t_1]$ and $E_{t_0-t_1}$ is a collection of links that were present at that moment. The goal of link prediction is to identify the set of linkages $E_{t'_0-t'_1}$ that exist between duration $[t'_0, t'_1]$ where $[t_0, t_1] \leq [t'_0, t'_1]$. Fig. 1

shows the structure of dynamic networks varying over timestamps. If $(t_1 = 1, t_2 = 2, \dots, t_n = n)$ are the timestamps and $(G_1, G_2, \dots, G_n)$ are the graphs of a snapshot of those timestamps. We need to predict G_{t+1} at $t+1$ timestamp based on observations from previous snapshots as potential links may become real links in dynamic networks with different snapshots [38]. In any snapshot of the network for a node x , $\Gamma(x)$ indicates the neighborhood of node x , and $k_x = |\Gamma(x)|$ denotes the number of neighbors of x . A is the adjacency matrix of graph $G(V, E)$ such that a_{ij} denotes the edge between i and j .

2.1. Network features extraction

Effective feature extraction is a crucial component of machine learning. These features can fall into various categories, such as topological similarity-based, probabilistic, node-specific semantic similarity-based, stochastic block model (SBM), Embedding-based, and more [9]. Among these, topological similarity-based features are particularly popular for several reasons: (1) Domain Independence: They are generic and do not rely on a specific domain. (2) Reflecting Network Behavior: Network features capture the topological characteristics of a network, which ultimately mirror the behavior of nodes within the network. This applies to a wide range of networks, whether they are social, biological, or commercial. Generally, topologically close entities tend to have more connections or interactions. For topological methods, feature extraction typically involves the computation of network metrics or indices. These indices are of different categories ranging from local, global, and quasi-local [7,8]. Some of the interesting attempts to utilize topological similarity are Path weight aggregation based link prediction (PWAFL) [39] where promising links are identified based on topological weights of path between two nodes and Cost-based feature for link prediction (CFLP) for estimating edge behavior throughout the entire network [40].

- **Local similarity indices** — Local similarity indices are computed utilizing information from the nodes' immediate surroundings. They can be easily extended to larger networks and have lower computational complexity because there are fewer nodes to take into account. However, we may need to examine their effectiveness because, in larger networks, their impact is diminished. For example, Common Neighbors (CN) [41] which prefers nodes having a higher common neighbor count, Adamic/Adar (AA) [42] where Common neighbor feature is penalized logarithmically by its appearance frequency, Jacquard Coefficient (JC) [43] normalizes common neighbor score by measuring the ratio of common neighbors by the complete set of neighbors, Preferential Attachment (PA) [44] here the basic assumption is a node links with higher probability to another node that has higher cardinality on a scale-free growing network.
- **Global similarity indices** — Indicators of Global similarity indices are computed utilizing a network's whole topological data. They typically include matrix-based operations on the entire network's adjacency matrix. As a result, these methods have greater computational complexity. Shortest Path (SP) [45] where the shortest path length is preferred, Random Walks (RW) [46] keeps on updating the probability vector of a node with node transition probability matrix on a random walk, Average Commute Time (ACT) [47] infers an average number of steps that a random walker starting from node x takes to reach a node y to and fro for the first time.
- **Quasi-local similarity indices** — Indexes of quasi-local similarity metrics are effective like local indices and seek to extract all of the network's topological data. They fall somewhere in the middle of local and global approaches, and many of their variations attempt to include a clustering-based strategy. Some of the popular network indices belonging to each category are local path Index (LP) [48] which sums the influence of all possible paths between two pairs of nodes, incrementally penalizing paths by their

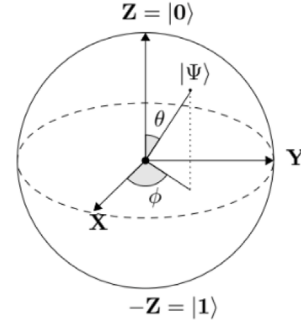


Fig. 2. Geometrical representation of qubit superposition.

length, Clustering Coefficient based Link Prediction (CCLP) [49] it calculates the clustering coefficient between the set of common neighbors, Path of Length 3 (L3) [50] which is Similar to the local path only paths of length 3 is considered.

2.2. Quantum computing basics

Quantum computing is a computation technology involving the development of systems and algorithms based on principles of Quantum theory. They have qubits that use superposition and gates that implement entanglement. Recent research has found that Quantum machine learning could take advantage of the exponentially huge quantum state space through controlled entanglement and interference, bringing speed and efficiency to the process [30,51]. Similar to how a classical computer is constructed from an electrical circuit of wires and logic gates, a quantum computer is constructed from a quantum circuit of wires and basic quantum gates that carry and modify quantum information [52].

2.2.1. Quantum representation

The basic building block of quantum computing is a Qubit – a quantum bit. Different from classical bits, a qubit can have multiple states at the same time. It is a two-level quantum system whose state can be described by complex numbers α and β

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (1)$$

where $\alpha^2 + \beta^2 = 1$. The two vectors $|0\rangle$ and $|1\rangle$ are orthonormal to each other. The qubit lies somewhere between these states and collapses to either of the two states when measured. The probabilities for $|0\rangle$ and $|1\rangle$ have the relation $p = |\alpha|^2$ and $1 - p = |\beta|^2$. Being a two-dimensional system, it makes sense to store the states of a qubit using two orthonormal state vectors as follows

$$|\psi\rangle = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \quad (2)$$

Also, note that this representation can also be parameterized by θ , we can use the trigonometric identity [52]: $\sqrt{\sin^2 \theta + \cos^2 \theta} = 1$ to describe the real α and β in terms of one variable, θ as $\alpha = \sin \frac{\theta}{2}$, $\beta = \cos \frac{\theta}{2}$

$$|\psi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle \quad (3)$$

This representation allows a geometric visualization of a qubit quantum state as a point on the surface of a unit radius sphere called the Bloch sphere [53]. Fig. 2 shows the representation of a qubit state on a Bloch sphere.

2.2.2. Quantum operations and gates

Any operations on these qubits can be performed through quantum gates which are unitary 2×2 matrices. Thus, having a property $UU^T =$

$U^T U = I$. Applying these operations on qubits results in vector-matrix multiplication. For example, applying a bit-flip gate denoted by $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ to the state $|\psi\rangle$

$$X|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix} = \beta|0\rangle + \alpha|1\rangle \quad (4)$$

The state of the two-qubit system $|\psi\rangle$ can be represented by using 4 complex coefficients giving the contribution of all possible classical states with a two-qubit system [54].

$$X|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle = \begin{pmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{pmatrix} \quad (5)$$

Also, the operations on these qubits are represented by a 4X4 unitary matrix. These are quantum supremacy gates as they perform entanglement. For example, a controlled-Z gate or CZ gate that adds a -1 phase to $|11\rangle$ basis.

$$CZ = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 \end{pmatrix}$$

which has a final effect on the two qubits as below: $CNOT|00\rangle = |00\rangle$; $CNOT|01\rangle = |01\rangle$;

$CNOT|10\rangle = |11\rangle$; $CNOT|11\rangle = |10\rangle$;

This gate is very crucial in quantum computing as Quantum computers are no more efficient than conventional computers without a 2-qubit gate like the CNOT as there would be no entanglement.

2.2.3. Superposition and entanglement

Superposition refers to the fact that any quantum state can be expressed as a linear combination of a few basis states. For instance, a single qubit can be represented using a linear combination of $|0\rangle$ and $|1\rangle$ as in Eq. (1). Similarly, the state of 'n' qubit system can be represented using 2^n bit string states (states formed by the tensor products of $|0\rangle$ and $|1\rangle$). The orthonormal basis formed by the 2^n bit-string states are called the computational basis.

But there can be situations for these qubits to be in a state that cannot be written as the tensor product of three single qubit states [55], for example,

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle) \quad (6)$$

Where it is only all 0s or 1s. States of a system that cannot be expressed as a tensor product of states of its individual subsystems are called entangled states. Entanglement means correlating that their individual state no longer exists. Thus, entanglement describes an essential property that is no longer sufficient to describe the state of each qubit separately; instead, you have to describe the probability of each combination of those qubits. So, for an 'n' qubit system we have to consider the probability of all 2^n states. This property makes quantum circuits difficult to simulate classically [56]. Also, it shows some of the incredible potentials of quantum computing. By passing a single qubit through a single logic gate we can perform 2^{n-1} calculations for a circuit of n qubits. For example, if we have 100 qubit systems passing a single qubit through a single logic gate, we perform $2^{99} = 6.33 \times 10^{29}$ i.e. 633 octillion calculations [57]. This is exponential speedup. Thus, entanglement makes it possible to create a complete 2^n dimensional complex vector space to do our computations using just 'n' physical qubits.

It is important to highlight that when measuring systems with multiple entangled qubits, the final measurement provides probabilities across various combined states. We interpret these results based on the state with the maximum magnitude and phase [58]. Keep in mind that measurement is a crucial step in the process of transitioning from quantum projection and optimization to retrieving results within the classical system

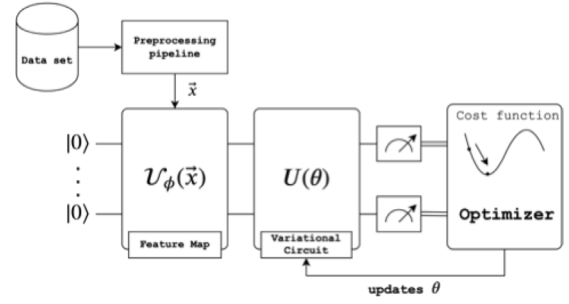


Fig. 3. Phases of Variational Quantum Classifier.

3. Quantum Machine Learning

Quantum Machine Learning (QML) encompasses the process of projecting classical data into quantum space, achieved through superposition via Hadamard Gates and entanglement facilitated by CNOT gates. Subsequently, QML involves training the data using quantum optimization and parameter shift techniques. Lastly, the quantum measurements use observable to convert the quantum probabilities to classical results [59,60]. In this part, the technical specifics of these stages are covered. Fig. 3 shows the overall structure of the phases of quantum classification. These phases are described in following subsections. Note that all of these operations are carried out in quantum space utilizing quantum circuits rather than traditional machines that operate on real scalar values [61].

3.1. Feature data quantum projection

The classical data first needs to be projected to complex quantum vector space using a parameterized feature map circuit, mathematically expressed by $\mathcal{U}_\phi(x)$ [30] as

$$\mathcal{U}_\phi(x) = \prod_d \bigcup_\phi (x) H^{\otimes n} \quad (7)$$

Where n is the number of qubits;

d is the depth of the circuit i.e. number of repetitions of:

1. Hadamard gate H and
2. Entangling block $U_{\phi(x)}$:

$$U_{\phi(x)} = \exp \left(i \sum_{s \subseteq [n]} \phi_s(x) \prod_{k \in s} P_k \right) \quad (8)$$

$P_i \in \{I, X, Y, Z\}$ denotes the Pauli matrices; $\phi_s(x)$ is the data mapping function; S describes the connectivity between qubits or data points: $S = \{ \binom{n}{k} \text{ combinations, } k=1, \dots, n \}$

As shown in Fig. 4 The H(Hadamard) gate projects the vectors to $|+\rangle$ and $|-\rangle$ serves the purpose of complex orthonormal basis and $U_{\phi(x)}$ performs entanglement and rotation along the Z axis, which makes the circuit hard to simulate classically [30], as it is a necessary condition to achieve quantum advantage.

3.2. Quantum training and classification

Parameterized quantum circuits, where the gates are defined through tunable parameters, are a fundamental building block of quantum machine learning algorithms. Classifiers in quantum machine learning basically encapsulate these parameterized circuits and use optimization techniques to quantify and eventually classify the circuit outcome. This can be visualized in Fig. 3, the encoded features after mapping to quantum space as explained in previous section are

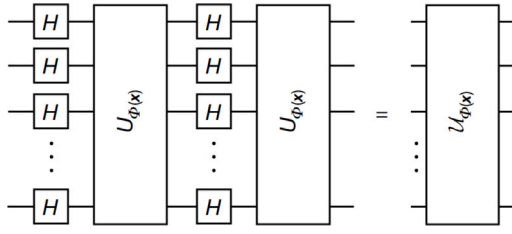


Fig. 4. Structure of Feature map

H represents the Hadamard gate and U represents the entanglement block, the repetition of these blocks result in U block.

passed through these parameterized circuits with set of initialization parameters, after the execution the circuit is measured with quantum measuring techniques. Based on the extracted labels the optimizers like SPSA and COBYLA compute the error gradients by applying the “parameter shift rule” [29,62]. Note that this is computed using the quantum circuits where it shifts the parameters up by a small amount and do the measurement and then down by a small amount and do the measurement and finally compute the difference of the two values. This is how the gradient of quantum circuit is calculated. Next the optimizer updates the parameter values of these variational circuit with the gradient value and repeats the whole process again. This is the loop similar to classical models that trains our parameters until the cost function value decreases. The final label extraction is done through quantum measurement when the model has converged. Note that the all the training related computations take place with quantum circuits be it gradient calculation, parameter updating or label extraction.

3.2.1. Variational quantum classifier (VQC)

It has a direct analogy with Support vector machines [30] one of the most popular classification methods in classical machine learning. It is basically part of the Qiskit Machine Learning library. As in the SVM, we project the data which is linearly inseparable to higher dimensional space so that it becomes linearly separable. Projecting the data to a quantum space have the in-built advantage of high dimensional space, hence this model after projecting the data using feature maps does nothing but linear classification. And to do so it needs a circuit with l -layers, parameterized by $\vec{\theta} \in R^{2n(l+1)}$ that will be optimized during the training. This is called the Variational/Parameterized quantum circuit Aka ansatz which is the core of Quantum machine learning. Notational representation of the quantum projection of features on a reference state $|0\rangle^n$, as done in the previous step can be given by $U_{\phi(x)}|0\rangle^n$ which is then passed to l -layer parameterized circuit $W(\vec{\theta})$ where $\vec{\theta} \in R^{2n(l+1)}$ that will be optimized during training. So, eventually, the circuit becomes

$$W(\vec{\theta})U_{\phi(x)}|0\rangle^n \quad (9)$$

The outcome of the circuit is a probability distribution over 2^n states and the next task is to identify the label from this outcome and optimize it, which will be covered in the next section.

3.2.2. Quantum neural networks (QNN) classifier

As classical neural networks are mapping from an input state to an output state. The correspondence between inputs and outputs is a function that is nothing but mapping inputs to desired output by optimizing the tunable network parameters. As in the classical neural network, a neuron may fire or not fire with 0 or 1 as output. For a Quantum neural network, the neuron’s quantum neural states are described by a two-dimensional Hilbert Space H_2 , spanned by the computational basis $B_2 = \{|0\rangle, |1\rangle\}$, where $|0\rangle$ encodes a non-firing neural state and $|1\rangle$ encodes a firing neural state. These physical descriptions of the states can be formulated as eigenstates of a neural firing Hamiltonian [63]:

$$\hat{H} = \frac{2\pi}{\tau} h \left(\frac{\hat{1} - \hat{\sigma}_3}{2} \right) \quad (10)$$

where τ can be measured in seconds, in order to produce the corresponding neural firing frequency as $(1/\tau)Hz$;

$$\hat{\sigma}_3 \text{ is Pauli's operator: } \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix};$$

The computational basis B_2 , will now, satisfy the eigenvalue equation:

$$\hat{H}|r\rangle = \frac{2\pi}{\tau} h|r\rangle \quad (11)$$

with $r = 0/1$. A neural network with N neurons has, thus, an associated Hilbert space, given by the N -tensor product of copies of H_2 : $H_2^{\otimes N}$, the basis $B_2^{\otimes N}$ spans the associated Hilbert Space: $B_2^{\otimes N} = \{|r\rangle : r \in A_2^N\}$ where A_2^N is the set of all length N binary strings:

$$A_2^N = \{r_1, r_2, \dots, r_N : r_k \in A_2, k = 1, 2, \dots, N\}$$

The basis $B_2^{\otimes N}$ represents the set of well-defined firing patterns for the QNN, which aligns with the classical states of ANN, though in general, the state of quantum computers exhibit a superposition of neural firing patterns represented by a normalized ket vector, in the space $H_2^{\otimes N}$. Hence the total energy of the firing neuron can be given as:

$$|\psi\rangle = \sum_{r \in A_2^N} \psi(r)|r\rangle \quad (12)$$

And the backpropagation operator $\hat{B}$ can be formulated as

$$\hat{B} = \sum_{r \in A_2^N} \hat{C}_s \otimes |S\rangle\langle S| \quad (13)$$

Where S represents the superposition-entangled state generated by the circuit C which has tunable parameters and s represents the parameter shift that would be performed. With the help of this architecture, quantum neural networks can behave as a type of self-programming quantum computer. They have two stages: the first stage being the neural learning interpretation stage, where the activation operator is applied, and the backpropagation stage, where the backpropagation operator is applied, continuously changing the parameter shift operator based on optimization until convergence is achieved.

4. Proposed work

The characteristic of nodes and edges emerging and disappearing over time is what defines dynamic networks [64,65]. This requires time-series forecasting, the literature [66,67] provides a complete overview of complex network techniques for nonlinear time series analysis. One static network for each time range can be thought of as making up the dynamic network [68]. For this, we divide the entire dataset into five chunks and considering the behavior in the previous four timestamps we predict the links in the last timestamp. For extracting the features, we chose a few network indices from each of the mentioned categories as listed in Section 2.1. We extract these features from each of the graphs generated at the four timestamps and consolidate them for subsequent input into the classifier. We have tested and validated our results on 5 standard dynamic network datasets. We provide a comparison between individual and combined performance indices, specifically focusing on three categories: local, global, and quasi-local indices. To evaluate the performance, we employ various metrics, including Accuracy, AUC, Precision, F1 score, among others. Detailed observations and explanations of the comparisons can be found in Section 5.

4.1. Parameterized quantum circuit based link prediction (PQCLP)

A model based on quantum techniques for predicting the behavior of a dynamic network and thus identifying the most promising missing links in the network. This basically lies in the category of probabilistic and statistical models, as quantum computation works on probabilities. In fact, taking the advantage of high dimensional probabilities due to

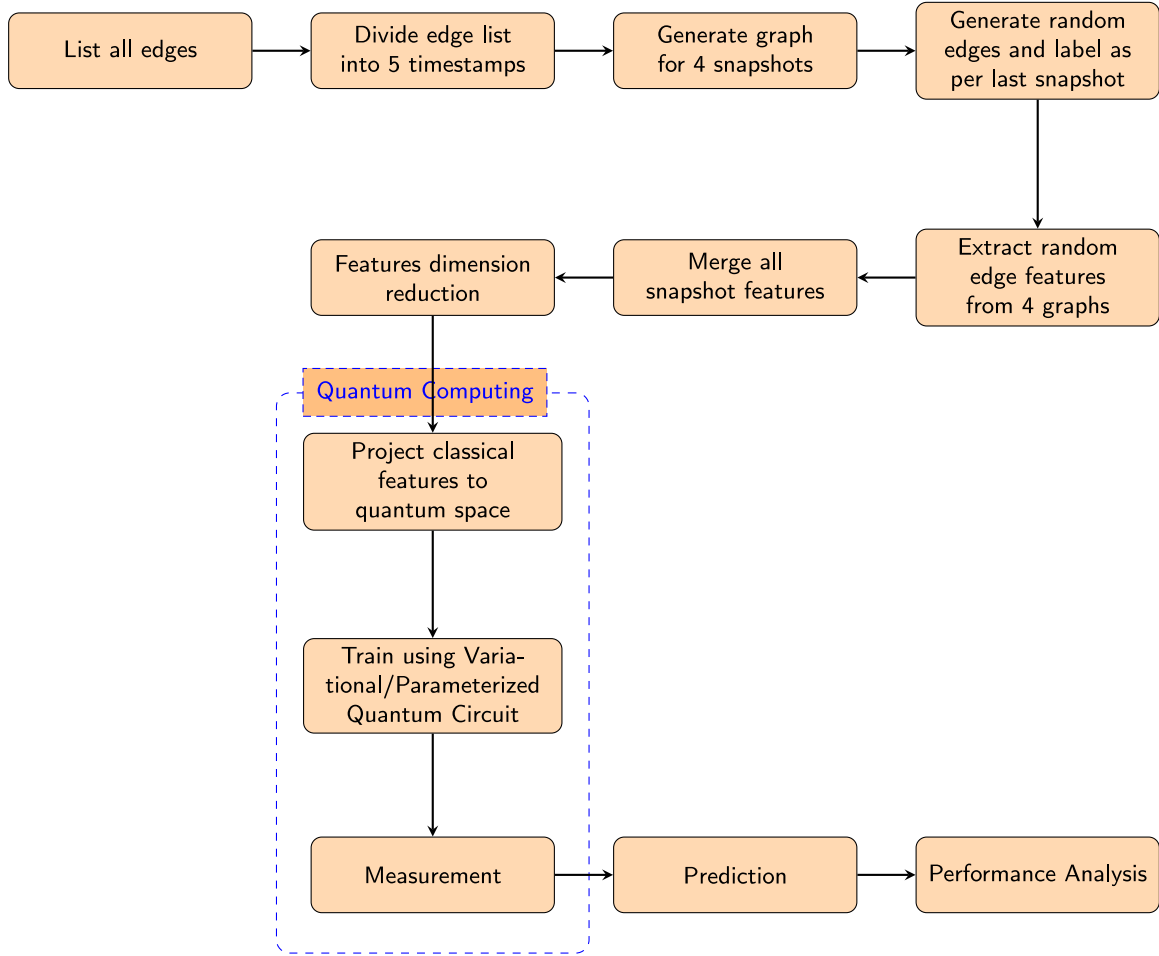


Fig. 5. Phases of PQCLP model.

inherent Hilbert space. It has phases spanning from classical feature extraction to quantum projection, quantum learning, and measurement then ultimately to classical result analysis and prediction. These are elaborated below and depicted in Fig. 5:

4.1.1. Feature extraction from dynamic network

- In our PQCLP model we consider dynamic datasets with time-lined link entries, the time next to the link indicates the time at which the link was present in the network. This value gives us an opportunity to partition the link entries in the dataset into different groups based on the time duration they belong to. Formally we create five snapshots of the whole network G as $(G_1, G_2 \dots G_5)$ for timestamps $(t_1, t_2 \dots t_n)$ with the condition that $t_i < t_{i+1}$.
- The idea here is that, based on the patterns in the previous snapshots predicting whether the link is present in the final snapshot or not. This leads us to use the first four snapshots for feature extraction and the last snapshot for validation and binary class labeling. The features set will be called $X = (x_1, x_2 \dots x_n)$ and class labels be called $Y = (y_1, y_2 \dots y_n)$
- Before proceeding with feature extraction, it is essential to ensure a balanced dataset for our model. Since we initially have only the true edges present in the last timestamp, we need to introduce false edges—edges that do not exist within the network at that timestamp. To achieve this balance, we generate a set of random non-existing edges from the nodes within the network. These generated edges, in equal quantity to the true edges, constitute our set of classification_edges. This step is crucial for effective training and evaluation of our model.

- The next step is to extract network features, the three of our model namely QNN_L/VQC_L have a combination of local topological indices i.e. Adamic Adar (AA), Preferential Attachment (PA), Jaccard Coefficient (JC), Common Neighbors (CN) whereas QNN_G/VQC_G has global topological indices i.e. Shortest Path (SP) and clustering based quasi-local i.e. Local Path (LP), LP with a path length of 3 (L3), clustering coefficient local path (CCLP) and finally QNN/VQC has a combination of all the feature from local, global category. The detailed algorithm for feature generation is explained through Algorithm 1 and explained in Section 4.2
- We basically iterate over the four snapshots, in every snapshot we iterate over edges present in classification_edges. For each edge, we try to get features for the model category i.e. local, global, or all and finally combine the features generated from each snapshot for the current edge. In the end, we append the class label based on the last snapshot i.e. 1 – if the edge is in the last snapshot and 0 – if it is from the false edges. Note that the dimension of the feature set from these combined network metrics will be $4 \times n$ where n is the number of metrics we choose for our execution.
- Having the features X and class labels Y ready we next divide the dataset into training and testing datasets, with a ratio of 8:2.
- Before providing the features to our quantum models, it is imperative to address the challenge of dimensionality. Quantum computers are fundamentally limited by the requirement that ‘ n ’ qubits demand storage and computational capacity of dimension 2^n in the case of simulators [56]. Hence, we use the Principal Component Analysis (PCA) for reducing the number of features at the cost of some loss of information. In case of individual

Algorithm 1: Feature Generation Algorithm for PQCLP**Input:** Dynamic social graph: $G(V,E)$, V : list of nodes, E : list of edges, $model_type \in \{local, global, all\}$, m number of partitions**Output:** features $||n||*||e||$: feature set for edges, n number of features, e number of edges

```

1   $edge\_FS \leftarrow dict$ 
2   $class\_label \leftarrow \{\}$ 
3   $G_{par}[] \leftarrow graph\_snapshot(G, m)$  ▷ Partition the graph in m snapshots
4   $E_m \leftarrow egdelist(G_{par}[m])$  ▷ last snapshot True Edges
5   $rnd\_false\_edges \leftarrow generate\_random\_false\_edges(G, len(E))$  ▷ False Edges
6   $E_{cl} \leftarrow sort(E_m + rnd\_false\_edges)$ 
7  for each snapshot in  $G_i \in G_{par}[m-1]$  do
8      for each  $E_i \in E_{cl}$  do
9           $node1, node2 \leftarrow edge\_nodes(E_i)$ 
10          $FeatureSet \leftarrow \{\}$ 
11         if  $E_i \in G_i$  then
12             if  $model\_type$  Is 'local' Or  $model\_type$  Is 'all' then
13                  $FeatureSet \leftarrow append\_local\_features(G_i, node1, node2)$  ▷ local features populated
14             end
15             if  $model\_type$  Is 'global' Or  $model\_type$  Is 'all' then
16                  $FeatureSet \leftarrow append\_global\_features(G_i, node1, node2)$  ▷ global features populated
17             end
18              $edge\_FS[E_i] \leftarrow append\_FeatureSet$ 
19         end
20         else
21              $edge\_FS[E_i] \leftarrow append \{0, 0, \dots\}$ 
22         end ▷ class labeling on last snapshot
23     if  $G_i$  Is  $G_{m-2}$  then
24         if  $E_i \in E_m$  then
25              $class\_label[i] = 1$ 
26         else
27              $class\_label[i] = 0$ 
28         end
29     end
30 end
31 return  $edge\_FS, class\_label$ 

```

categories local/global, 16 features generated from 4 snapshots are reduced to 4 and for combined category including both local and global, 32 features from 4 snapshots are reduced to 8 features.

4.1.2. Quantum-based training and evaluation

Post extracting the features from the dynamic network we move toward the quantum part of our experiment. Here we use two different quantum classifiers namely VQC and QNN which have an analogy with their classical counterparts namely Support Vector Machines and Quantum Neural Networks in terms of training and optimizing the parameters. Broadly the training phase includes the following steps:

- **Feature data Quantum projection:** The first step is to convert the classical data to quantum bits i.e. qubits and map them to complex vector quantum space with a quantum feature map. As discussed in Section 4.1 this requires repetitions of the Hadamard gate and entangling blocks based on the depth and entanglement criteria specified. We choose to keep the circuit depth 2 with entanglement linear for sake of simplicity. The circuit for the phase can be visualized with Fig. 6
- **Quantum training and parameter tuning:** For training and weight adjustments we have variational circuits aka Ansatz. As discussed in Section 4.2 they are nothing but parameterized quantum circuits, which contain alternating layers of parameterized rotations and entanglement. Here we create Ansatz with a depth of 2 and use Y and Z rotations with Ry and Rz gates as in Fig. 7

- **Quantum Optimization:** With the Ansatz, fine-tuned we are ready with the circuit that can fine-tune the parameters to reduce the loss/cost function and improve the accuracy of the results. But just like classical systems this also needs to be channelized with proper optimization algorithms like Gradient Decent. Real quantum machines are noisy and observations are corrupted by noise. It has been observed in [69] that stochastic approximation algorithm performs well in the noisy experimental setting, hence for quantum run-time on IBM quantum device i.e. ibm_oslo we choose to use Simultaneous Perturbation Stochastic Approximation (SPSA), which has a gradient-free approach and does not require gradient at the runtime. It only needs the cost function values. Whereas for classical simulation that are noise free, we found COBYLA, which is a Constrained Optimizer using Linear Approximation [70], perform better.
- **Quantum Classifier:** We choose to use two variants of quantum classifiers Variational Quantum Classifier (VQC) and Quantum Neural Network Classifier (QNN) explained in Section 3.2 in detail. Both these classifiers perform the task of classification with these inputs provided to them (1) a feature map (2) a quantum instance specifying the back-end used i.e. whether a local simulator or IBM run-time simulator or real IBM quantum machine (3) parameterized circuits aka Ansatz. VQC classifier tries to find the separating hyperplane and the maximum support vectors whereas the QNN classifier having neural network architecture

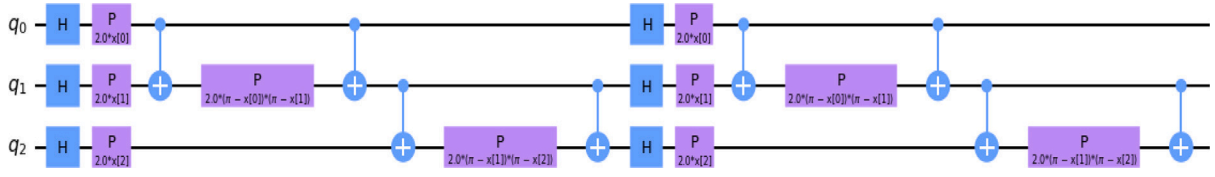


Fig. 6. Circuit Diagram obtained from IBM_OSLO real quantum computer for ZZ Feature-map of QNN and VQC. Blue H blocks represent Hadamard Gates, The small purple P blocks represent the Pauli Z gates working on single qubit and large purple P blocks represent Pauli ZZ gates working on two qubit entanglement. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

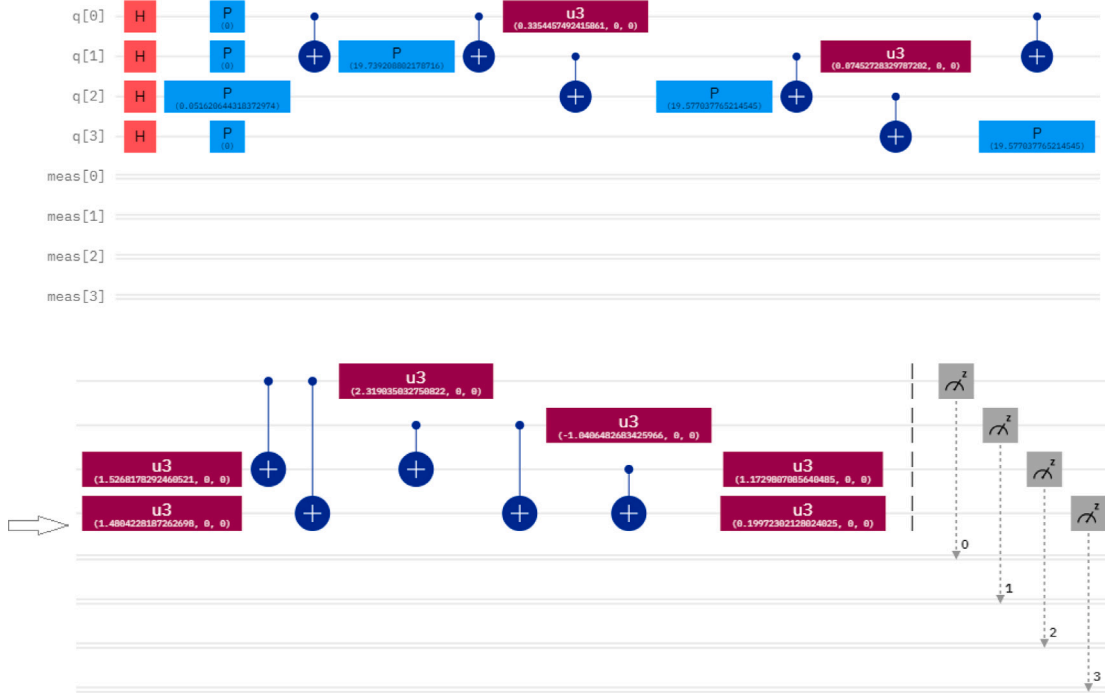


Fig. 7. Circuit Diagram from IBM_OSLO real quantum computer for Variational Circuit, Red H blocks represent Hadamard Gates, the small blue P blocks represent the Pauli Z gate working on single qubit, large blue P blocks represent Pauli ZZ gates working on two qubit entanglement and the maroon U3 blocks represent the parameterized rotational gates. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

has a forward activation and backpropagation optimization mechanism. With these algorithms both the classifiers perform binary classification following quantum measurement.

- **Quantum Measurement:** In this phase, after the convergence in measurements, the quantum bits will finally collapse to one of the basis states and final classification will be done on classical bits i.e. 0 or 1 in our case, with 1 indicating link presence and 0 indicating absence while training and testing.

4.2. Algorithm for feature generation

Algorithm [1] explains in detail the process of feature extraction from the dynamic network before propagation to the Quantum Circuit for projection to Quantum space. This has been roughly touched while explaining the steps of the flowchart. The algorithm begins with the initialization of necessary empty variables in lines 1 & 2 namely *edge_FS*: an empty dictionary to store edge as key and its features as values, and *class_label*: an array to store 1/0 for indicating the presence or absence of the edge in the last snapshot. In line 3 *G_{par}* stores the partitions of the whole graph, basically the edges divided based on the timestamps next to it. In line 4 *E_m* stores the edges of the last snapshot. We generate a set of random false edges with the function *generate_random_false_edges* and merge these *rnd_false_edges* with *E_m* to form a balanced set of classification edges *E_{cl}* in the lines 5 & 6. Now we iterate over all the graph partitions and while iterating over these partitions we next

iterate over all the classification edges *E_{cl}* in the next two lines 7 & 8. In lines 9–22, for each edge, if the edge in the *E_{cl}* belongs to the current graph *G_i*, we extract the two endpoints i.e. *node1* and *node2*. With the help of these nodes we extract features for the edge based on the model category it belongs to i.e. if it is a local model extract *AA*, *CN*, *PA*, *JC* using *local_features* function and if it is a global model extract *SP*, *L3*, *LP*, *CCLP* using *global_features* function and all of these if it is a combined model. Else if the edge does not belong to the current graph we just append a set of zeros as many as the number of features for that model category in lines 23 & 24. This iteration takes place till the second last partition and makes the task of feature extraction complete and in the last partition iteration we use the partition for class labeling i.e. in lines 25–27, if the edge in *E_{cl}* is present in the last partition *G_m*, we mark the *class_label* as 1 and 0 otherwise. Finally, our algorithm returns the set of features *edge_FS* and *class_label* in the last line 28 for further processing to our Quantum Model.

4.3. Prediction and performance evaluation

Once our model is trained, we finally perform classification and prediction on the test data. We try to evaluate several performance metrics with the results obtained and present a comparison between these outcomes in the next section. Note that we perform ten iterations for each graph, each time having the same snapshot structure with real edges but every time generating different random false edges and then

take the average of scores of all the ten iterations. This gives a slightly different set of classification edges each time for the same graph, and as a result, we have even more concrete and general probabilities as well as evaluation scores. With the predicted values 1 – for the existing edge and 0 – for the non-existing edge, for classification edges, we predict the accuracy of the model by evaluating the ratio of correct prediction using several performance metrics discussed in the next section.

4.4. Algorithm optimization

The algorithms exhibit a tendency toward reduced computational efficiency when executed on quantum simulators, and when employed on genuine quantum hardware, they are constrained by the limited number of qubits and an exponentially growing parameter count attributed to entangled qubits. To mitigate these challenges, a potential solution lies in the customization of the algorithm to leverage the advantages offered by the kernel trick similar to classical machine learning models. The introduction of Quantum Neural Tangent Kernels (QNTK) [71,72] facilitates the exploration of high-dimensional spaces through a simplified kernel function. Notably, when these QNTK are applied within the framework of lazy training, the residual error exhibits an exponential decay. This phenomenon occurs as the QNTK effectively captures the major principal components and the direction in which the convergence is faster. This strategic adjustment promises to yield an exponential acceleration in the convergence process, substantially reducing the time required for quantum computation and simulation during each cycle.

Additionally, for scenarios involving a substantial number of qubits, the integration of Tensor Networks (TN) [73,74] proves highly advantageous. Tensor networks provide a robust method for representing systems of entangled qubits as interconnected networks of tensors, with the number of physical qubits scaling logarithmically or independently in relation to input and output data sizes. The algorithm can be further enhanced by adapting it to convert multiple inputs into the tensor network representation. This representation can be refined through optimization techniques that consider entanglement entropy, resulting in a compact encoding of data into low-qubit states. Particularly promising tensor network structures, such as Matrix Product States (MPS) [75] or Tree Tensor Networks (TTN) [76], can be employed to effectively implement this approach.

5. Experimental setup

In this section, we narrate the different runtime environments, datasets, and several performance metrics used to evaluate the proposed models *PQCLP-QNN* and *PQCLP-VQC*. Subsequently, presenting the analysis of the performance of our models on different feature combinations, test ratios, and different comparable models. We decide to use smaller and moderate-size datasets for our tests because the computation time on the simulator increases with larger datasets. As discussed in Section 4.1.2, the steps to build a quantum classifier involve the choice of function for (1) Feature Projection (2) ansatz/ parameterized circuit (3) Optimizers (4) Classifiers and (5) Runtime Environment. We use ZZFeatureMap for feature projection from Qiskit circuit library as it gives a genuine entanglement block of 2, with layers of Hadamard gates, capable to project the input to complex space that is hard to simulate classically. For training the Ansatz/Parameterized circuit that we use is RealAmplitudes from Qiskit circuit library. It has single qubit rotation followed by 2 qubit entanglement layers. This configuration gives us genuine results without increasing the execution complexity. For optimizer as discussed previously for real quantum machines SPSA does better and for classical simulations COBYLA performs well. The classifier for two variations of our algorithm are Variation Quantum Classifier (VQC) and Neural Network Classifier (QNN) from Qiskit Machine Learning library. The run-time options we explored include

quantum simulator on a local machine using Qiskit Circuit library-0.24.0 and Qiskit machine learning library-0.6.0, a quantum simulator in IBM Quantum run-time environment i.e. *ibmq_qasm_simulator* (having 32 qubits capacity). Also, we got the opportunity to run very few instances on the real quantum machine from IBM Quantum runtime i.e. *ibmq_oslo* (having 7 Qubits with Falcon r5.11H processor having 2.6 K CLOPS capacity).

5.1. Evaluation metrics

In order to analyze the model performance we use six performance evaluation matrices i.e. Accuracy — the number of correct predictions w.r.t overall predictions [77], Area under the precision–recall curve (AUPR) — an average of precision across all recall values [78], Area under the ROC curve (AUROC/AUC) — the likelihood that a non-observed but existing link will have a higher score than other truly nonexistent relationships [79], Precision — a measure of how many of the positive predictions made are correct [80], Recall/Sensitivity — a measure of how many of the positive cases the classifier correctly predicted, over all the positive cases in the data [81], F1 score — a trade-off between precision and recall [82].

5.2. Datasets

We have evaluated our results on 5 well-known standard datasets for dynamic link prediction. The two datasets MIT [83] and Radoslaw-email [84] may be considered dense datasets or densely connected networks having (96 nodes, 10,86,404 edges) and (167 nodes and 82,876 edges) respectively, as both have fewer nodes and a large number of edges. Eu-core [85] may be considered medium density dataset with 986 nodes, and 3,32,334 edges. Whereas Fb-forum [86] and CollegeMsg [87] can be termed low-density dataset having (899 nodes and 33,686 edges) and (1899 nodes and 59,835 edges) respectively. We demonstrate our results on each type of dataset ranging from small to large and sparse to dense networks.

6. Result analysis

We conduct some experiments to observe whether our model follows the right learning pattern i.e. when it is provided with more features whether it is able to predict better. We plan to use different feature categories one by one, in order to watch the impact of each category on the performance. Next, we also try to capture the performance of our quantum models over different train-test ratios for critical metrics like accuracy and precision. We also compare our quantum models with their classical counterparts, categorized separately. Furthermore, we assess our models against popular state-of-the-art models using various performance metrics.

6.1. Comparison of PQCLP models on cumulative features with individual features from local category

Initially, We conduct a comparison between the individual features of local categories with the performance of combined features of that category on our quantum models. So basically, a combination of 4 local features like Adamic Adar (AA), Common Neighbor(CN), Jacquard Coefficient (JC), and Preferential Attachment (PA) from 4 snapshots, eventually 16 features are passed to the proposed *PQCLP-QNN* and *PQCLP-VQC* models and then these models are provided each of these individual features from 4 snapshots, eventually 4 features and evaluated performance is depicted in Tables 1 and 2. We use a 4-qubit quantum classifier with a stochastic SPSA optimizer with 100 epochs and a learning rate of 0.001 for both quantum models. On a detailed analysis of scores in Table 1 we observe that for Accuracy, AUPR, and F1 score evaluation *PQCLP-QNN_Local* performs better than all the individual features for radoslaw-email, eu-core, and MIT and

Table 1Performance comparison and analysis of proposed *PQCLP-QNN* with individual local features.

Metric	Dataset	AA	CN	JC	PA	PQCLP-QNN_L
Test Accuracy	fb-forum	0.5928	0.5828	0.5438	0.4728	0.5572
	radoslaw-email	0.6828	0.7111	0.6225	0.5337	0.7697
	Eu-core	0.7230	0.7062	0.7389	0.4936	0.7678
	CollegeMsg	0.5299	0.5706	0.5431	0.4883	0.5503
	MIT	0.6196	0.6058	0.5705	0.5341	0.7677
AUC	fb-forum	0.6014	0.5882	0.5562	0.5000	0.5606
	radoslaw-email	0.6956	0.7198	0.6297	0.5000	0.8275
	Eu-core	0.7209	0.7146	0.7409	0.5055	0.7679
	CollegeMsg	0.5334	0.5644	0.5504	0.5048	0.5540
	MIT	0.6255	0.6161	0.5718	0.5000	0.7718
PRECISION	fb-forum	0.7467	0.7208	0.6242	0.2417	0.8257
	radoslaw-email	0.8282	0.8177	0.7023	0.5337	0.8134
	Eu-core	0.9683	0.9554	0.9469	0.4496	0.9502
	CollegeMsg	0.5788	0.6474	0.7092	0.2973	0.7826
	MIT	0.6855	0.6816	0.5978	0.5341	0.8077
RECALL	fb-forum	0.3230	0.2927	0.5223	0.5000	0.1562
	radoslaw-email	0.5163	0.5883	0.5196	1.0000	0.7689
	Eu-core	0.4570	0.4512	0.5111	0.8065	0.5693
	CollegeMsg	0.4538	0.2652	0.1801	0.5530	0.1495
	MIT	0.5318	0.4601	0.5012	1.0000	0.7562
AUPR	fb-forum	0.7151	0.6848	0.6911	0.7553	0.7063
	radoslaw-email	0.8011	0.8132	0.7360	0.7668	0.8511
	Eu-core	0.8485	0.8428	0.8545	0.7250	0.8692
	CollegeMsg	0.6432	0.6364	0.6529	0.7390	0.6714
	MIT	0.7451	0.7103	0.6762	0.7671	0.8465
F1 MEASURE	fb-forum	0.4471	0.4066	0.4982	0.6754	0.2572
	radoslaw-email	0.6347	0.6844	0.5889	0.6958	0.7800
	Eu-core	0.6225	0.6063	0.6678	0.6304	0.7097
	CollegeMsg	0.4274	0.3713	0.2788	0.6510	0.2461
	MIT	0.6104	0.5471	0.5421	0.6958	0.7732

Here the proposed model *PQCLP-QNN* first gets 4 features from local category from 4 snapshots, a total of 16 features, and the rest of the models are provided each feature from 4 snapshots i.e. 4 features.

Table 2Performance comparison and analysis of proposed *PQCLP-VQC* with individual local features.

Metric	Dataset	AA	CN	JC	PA	PQCLP-VQC_L
Test Accuracy	fb-forum	0.6299	0.6388	0.6297	0.4886	0.5674
	radoslaw-email	0.7036	0.6704	0.6093	0.5304	0.8050
	Eu-core	0.8138	0.8310	0.7972	0.4810	0.8351
	CollegeMsg	0.5848	0.5665	0.5904	0.4787	0.5756
	MIT	0.7363	0.7324	0.7144	0.5401	0.7411
AUC	fb-forum	0.6293	0.6381	0.6294	0.5000	0.5691
	radoslaw-email	0.7047	0.6678	0.6079	0.5000	0.8082
	Eu-core	0.8109	0.8319	0.7957	0.5000	0.8349
	CollegeMsg	0.5812	0.5670	0.5909	0.5000	0.5851
	MIT	0.7441	0.7334	0.7125	0.5000	0.7441
PRECISION	fb-forum	0.6739	0.6838	0.6993	0.2953	0.7746
	radoslaw-email	0.7367	0.6841	0.6173	0.5304	0.8562
	Eu-core	0.8824	0.8960	0.8371	0.2405	0.9625
	CollegeMsg	0.7355	0.6648	0.7298	0.1447	0.8032
	MIT	0.8125	0.7636	0.7211	0.5401	0.8056
RECALL	fb-forum	0.4915	0.6327	0.4545	0.6000	0.1984
	radoslaw-email	0.6818	0.7076	0.6137	1.0000	0.8839
	Eu-core	0.7118	0.7534	0.7256	0.5000	0.6962
	CollegeMsg	0.2666	0.2756	0.2904	0.3000	0.5718
	MIT	0.6845	0.6859	0.7493	1.0000	0.6136
AUPR	fb-forum	0.7091	0.5191	0.7130	0.7510	0.6877
	radoslaw-email	0.7931	0.7737	0.7167	0.7652	0.8716
	Eu-core	0.8671	0.8870	0.8487	0.7499	0.9051
	CollegeMsg	0.6845	0.6519	0.6878	0.7095	0.7553
	MIT	0.8361	0.8041	0.8013	0.7700	0.8139
F1 MEASURE	fb-forum	0.6209	0.6381	0.6165	0.3282	0.4987
	radoslaw-email	0.7031	0.6678	0.6037	0.3465	0.8292
	Eu-core	0.8107	0.8300	0.7949	0.3248	0.8313
	CollegeMsg	0.5377	0.5253	0.5093	0.3237	0.5143
	MIT	0.7055	0.7209	0.7117	0.3503	0.7263

Here the proposed model *PQCLP-VQC* first gets 4 features from local category from 4 snapshots, a total of 16 features, and the rest of the models are provided each feature from 4 snapshots i.e. 4 features.

Table 3Performance comparison and analysis of proposed *PQCLP-QNN* with individual global features.

Metric	Dataset	L3	LP	CCLP	SP	PQCLP-QNN_G
Test Accuracy	fb-forum	0.7019	0.5374	0.5899	0.6610	0.7335
	radoslaw-email	0.6926	0.6462	0.6644	0.4767	0.7816
	Eu-core	0.7347	0.7208	0.6886	0.3575	0.7891
	CollegeMsg	0.6320	0.5066	0.5543	0.5995	0.6234
	MIT	0.6062	0.5737	0.6266	0.4647	0.6397
AUC	fb-forum	0.6992	0.5495	0.5728	0.6618	0.7300
	radoslaw-email	0.7008	0.6565	0.6832	0.4888	0.8342
	Eu-core	0.7353	0.7268	0.6910	0.3576	0.7912
	CollegeMsg	0.6379	0.5250	0.5619	0.5909	0.6598
	MIT	0.6103	0.5809	0.6305	0.4791	0.6442
PRECISION	fb-forum	0.9873	0.8285	0.9165	0.7277	0.9559
	radoslaw-email	0.7899	0.7658	0.8584	0.4518	0.8168
	Eu-core	0.9654	0.9569	0.9641	0.2571	0.9657
	CollegeMsg	0.9087	0.7697	0.8976	0.7798	0.9744
	MIT	0.6748	0.6365	0.7242	0.4555	0.7217
RECALL	fb-forum	0.4041	0.2242	0.1606	0.5263	0.4926
	radoslaw-email	0.5775	0.5446	0.4551	0.2741	0.8443
	Eu-core	0.4881	0.4762	0.3966	0.2367	0.8815
	CollegeMsg	0.3071	0.2792	0.1417	0.2982	0.5670
	MIT	0.5122	0.4664	0.4822	0.2696	0.7371
AUPR	fb-forum	0.8415	0.7230	0.7371	0.7440	0.8490
	radoslaw-email	0.7964	0.7796	0.8070	0.5541	0.8681
	Eu-core	0.8543	0.8532	0.8326	0.4377	0.7810
	CollegeMsg	0.7835	0.7130	0.7396	0.7160	0.7971
	MIT	0.7188	0.6926	0.7409	0.5631	0.6182
F1 MEASURE	fb-forum	0.5681	0.2704	0.2731	0.6062	0.6405
	radoslaw-email	0.6682	0.6257	0.5977	0.3298	0.7337
	Eu-core	0.6469	0.6376	0.5602	0.2312	0.6133
	CollegeMsg	0.4600	0.2666	0.2425	0.4046	0.7100
	MIT	0.5753	0.5349	0.5771	0.3302	0.4976

Here the proposed model *PQCLP-QNN* first gets 4 features from global category from 4 snapshots, a total of 16 features, and the rest of the models are provided each feature from 4 snapshots i.e. 4 features.

marginally misses the best performer in case of fb-forum and collegemsg. It has the best AUC score on all other datasets except fb-forum. Similarly, for a Precision score, it performs best on all other datasets except radoslaw-email and eu-core, where it slightly misses the best score. Considering the recall scores, it has low performance, primarily as a consequence of low density i.e. a sparse network, resulting in a large number of false negatives. Similarly, on detailed analysis of scores in Table 2, *PQCLP-VQC_Local* has better Accuracy, AUC, and F1 score than all the individual features for radoslaw-email, eu-core, and MIT and marginally misses the best performer in case of fb-forum and collegemsg. For recall having moderate performance on most of the datasets. Lastly, it has the best precision and AUPR score on most of the datasets except fb-forum. Hence the model behaves in the right way i.e. prediction scores increase when the model receives more combinations of features.

6.2. Comparison of *PQCLP* models on cumulative features with individual features from global category

Similarly, Tables 3 and 4 compare performances of all combined global features with individual global features on *PQCLP-QNN* and *PQCLP-VQC* models respectively. The two models are initially provided 4 global features such as Shortest Path (SP), local path Index (LP), Coefficient based Link Prediction (CCLP), and Path of Length 3 (L3) from 4 snapshots, so eventually 16 features, and then they are fed each of these features individually from 4 snapshots, hence 4 features. We use a 4-qubit quantum classifier with a stochastic SPSSA optimizer with 100 epochs and a learning rate of 0.001 for both quantum models. It is evident from the Table 3 that *PQCLP-QNN_Global* with all global features outperforms the individual performance of SP, L3, LP, and CCLP in terms of Test accuracy and AUC for all the datasets. Unlike the local features, we have a good performance for global features in terms of recall. For precision again, the model is touching or is the best

performer for all the datasets. It has the best AUPR and F1 score on fb-forum, radoslaw-email, and collegemsg and moderate on eu-core but low on MIT. Now analyzing the Table 4 Our *PQCLP-VQC_Global* model outperforms all the datasets in terms of precision and AUPR. Whereas the Accuracy, AUC, and F1 score for *PQCLP-VQC_Global* are best for fb-forum, radoslaw-email, and MIT and nearly touch the best score in eu-core and collegemsg. Recall scores for eu-core and collegemsg are low for the reasons mentioned in the previous comparison, moderate on MIT and best on fb-forum and radoslaw-email. We may observe that having global features with better performance our global models perform better than the local models.

6.3. Comparison of *PQCLP* models on cumulative features with individual features from both local and global category

Next, we move to combined features from both the categories of network metrics i.e. local, global, and quasi-local. Basically, the proposed models *PQCLP-QNN_LG* and *PQCLP-VQC_LG* in this category are provided a combination of features namely AA, CN, JC, PA, SP, L3, LP, and CCLP from 4 snapshots, eventually, 32 features, next they are provided each of the mentioned features one by one from 4 snapshots that results in 4 features and the comparison is presented in the Tables 5 and 6. We use 8 qubit quantum classifiers with a stochastic SPSSA optimizer with 50 epochs and a learning rate of 0.001 for both quantum models. These tables show the comparison between the performance of individual metrics with the combined performance of all these features on both the proposed quantum models namely *PQCLP-QNN* and *PQCLP-VQC*. On a detailed analysis of the Table 5, we observe that AUC and accuracy are best on all the other datasets except fb-forum where it is moderate. It has the best Precision and AUPR score on radoslaw-email, eu-core, and MIT but low on collegemsg and fb-forum. It has low Recall scores for eu-core, collegemsg, and MIT but best scores for radoslaw-email and fb-forum. F1 score is best on radoslaw-email, eu-core, and collegemsg,

Table 4Performance comparison and analysis of proposed *PQCLP-VQC* with individual global features.

Metric	Dataset	L3	LP	CCLP	SP	PQCLP-VQC_G
Test Accuracy	fb-forum	0.7259	0.6181	0.6432	0.5881	0.7724
	radoslaw-email	0.6978	0.7698	0.7014	0.3826	0.8141
	Eu-core	0.8347	0.8350	0.8157	0.4107	0.8250
	CollegeMsg	0.6853	0.5345	0.5635	0.6249	0.6609
	MIT	0.7442	0.6061	0.7648	0.3924	0.7798
AUC	fb-forum	0.7305	0.6215	0.6470	0.5912	0.7660
	radoslaw-email	0.6994	0.7694	0.7038	0.4101	0.8142
	Eu-core	0.8339	0.8348	0.8116	0.4167	0.8226
	CollegeMsg	0.6799	0.5464	0.5636	0.6231	0.6627
	MIT	0.7465	0.6210	0.7654	0.4139	0.7802
PRECISION	fb-forum	0.8888	0.7847	0.8243	0.6409	0.9744
	radoslaw-email	0.7280	0.7911	0.7459	0.2887	0.8675
	Eu-core	0.8807	0.9410	0.9081	0.3414	0.9749
	CollegeMsg	0.8263	0.7395	0.7969	0.6252	0.9509
	MIT	0.7892	0.7207	0.8152	0.3336	0.8573
RECALL	fb-forum	0.5310	0.3441	0.3769	0.4877	0.5453
	radoslaw-email	0.6892	0.7747	0.6751	0.0940	0.6906
	Eu-core	0.7699	0.7141	0.6890	0.2160	0.6621
	CollegeMsg	0.4498	0.1598	0.1732	0.6744	0.3439
	MIT	0.6957	0.4750	0.6931	0.1368	0.6726
AUPR	fb-forum	0.8300	0.7297	0.7584	0.6950	0.8702
	radoslaw-email	0.7909	0.8433	0.7981	0.4372	0.8623
	Eu-core	0.8822	0.8989	0.8738	0.4803	0.9017
	CollegeMsg	0.7723	0.6662	0.6916	0.7320	0.8124
	MIT	0.8222	0.7424	0.8317	0.4660	0.8470
F1 MEASURE	fb-forum	0.7154	0.5843	0.6161	0.5802	0.7578
	radoslaw-email	0.6969	0.7684	0.7010	0.3293	0.8361
	Eu-core	0.8335	0.8322	0.8112	0.3822	0.8191
	CollegeMsg	0.6644	0.4600	0.4835	0.6194	0.6228
	MIT	0.7424	0.6015	0.7631	0.3474	0.7761

Here the proposed model *PQCLP-VQC* first gets 4 features from global category from 4 snapshots, a total of 16 features, and the rest of the models are provided each feature from 4 snapshots i.e. 4 features.

Table 5Performance comparison and analysis of proposed *PQCLP-QNN* with individual features from both local and global category.

Metric	Dataset	AA	CN	JC	PA	SP	L3	CCLP	CCLP2	PQCLP-QNN_LG
Test Accuracy	fb-forum	0.5928	0.5828	0.5438	0.4728	0.6610	0.7019	0.5374	0.5899	0.6806
	radoslaw-email	0.6828	0.7111	0.6225	0.5337	0.4767	0.6926	0.6462	0.6644	0.7935
	Eu-core	0.7230	0.7062	0.7389	0.4936	0.3575	0.7347	0.7208	0.6886	0.8091
	CollegeMsg	0.5299	0.5706	0.5431	0.4883	0.5995	0.6320	0.5066	0.5543	0.6301
	MIT	0.6196	0.6058	0.5705	0.5341	0.4647	0.6062	0.5737	0.6266	0.6797
AUC	fb-forum	0.6014	0.5882	0.5562	0.5000	0.6618	0.6992	0.5495	0.5728	0.6722
	radoslaw-email	0.6956	0.7198	0.6297	0.5000	0.4888	0.7008	0.6565	0.6832	0.8993
	Eu-core	0.7209	0.7146	0.7409	0.5055	0.3576	0.7353	0.7268	0.6910	0.7912
	CollegeMsg	0.5334	0.5644	0.5504	0.5048	0.5909	0.6379	0.5250	0.5619	0.6121
	MIT	0.6255	0.6161	0.5718	0.5000	0.4791	0.6103	0.5809	0.6305	0.6442
PRECISION	fb-forum	0.7467	0.7208	0.6242	0.2417	0.7277	0.9873	0.8285	0.9165	0.7875
	radoslaw-email	0.8282	0.8177	0.7023	0.5337	0.4518	0.7899	0.7658	0.8484	0.8496
	Eu-core	0.9683	0.9554	0.9469	0.4496	0.2571	0.9600	0.9569	0.9641	0.9603
	CollegeMsg	0.5788	0.6474	0.7092	0.2973	0.7798	0.9087	0.7697	0.8976	0.7881
	MIT	0.6855	0.6816	0.5978	0.5341	0.4555	0.6748	0.6365	0.7242	0.7218
RECALL	fb-forum	0.3230	0.2927	0.5223	0.5000	0.5263	0.4041	0.2242	0.1606	0.6187
	radoslaw-email	0.5163	0.5883	0.5196	1.0000	0.2741	0.5775	0.5446	0.4551	0.7337
	Eu-core	0.4570	0.4512	0.5111	0.8065	0.2367	0.4881	0.4762	0.3966	0.7133
	CollegeMsg	0.4538	0.2652	0.1801	0.5530	0.2982	0.3071	0.2792	0.1417	0.4256
	MIT	0.5318	0.4601	0.5012	1.0000	0.2696	0.5122	0.4664	0.4822	0.4976
AUPR	fb-forum	0.7151	0.6848	0.6911	0.7553	0.7440	0.8415	0.7230	0.7371	0.7642
	radoslaw-email	0.8011	0.8132	0.7360	0.7668	0.5541	0.7964	0.7796	0.8070	0.8833
	Eu-core	0.8485	0.8428	0.8545	0.7250	0.4377	0.8543	0.8532	0.8326	0.8815
	CollegeMsg	0.6432	0.6364	0.6529	0.7390	0.7160	0.7835	0.7130	0.7396	0.6915
	MIT	0.7251	0.7103	0.6762	0.7671	0.5631	0.7188	0.6926	0.7209	0.7371
F1 SCORE	fb-forum	0.4471	0.4066	0.4982	0.6754	0.6062	0.5681	0.2704	0.2731	0.6505
	radoslaw-email	0.6347	0.6844	0.5889	0.6958	0.3298	0.6682	0.6257	0.5977	0.8128
	Eu-core	0.6225	0.6063	0.6678	0.6304	0.2312	0.6469	0.6376	0.5602	0.7461
	CollegeMsg	0.4274	0.3713	0.2788	0.6510	0.4046	0.4600	0.2666	0.2425	0.4951
	MIT	0.6104	0.5471	0.5421	0.6958	0.3302	0.5753	0.5349	0.5771	0.5817

Here the proposed model *PQCLP-QNN* first gets 8 features from local and global categories from 4 snapshots, a total of 32 features, and the rest of the models are provided each feature from 4 snapshots i.e. 4 features.

Table 6Performance comparison and analysis of proposed *PQCLP-VQC* with individual features from both local and global category.

Metric	Dataset	AA	CN	JC	PA	SP	L3	CCLP	CCLP2	PQCLP-VQC_LG
Accuracy	fb-forum	0.6299	0.6317	0.6297	0.4886	0.5881	0.7259	0.6181	0.6432	0.6579
	radoslaw-email	0.7036	0.6704	0.6093	0.5304	0.3826	0.6978	0.7698	0.7014	0.8196
	Eu-core	0.8138	0.8310	0.7972	0.4810	0.4107	0.8347	0.8350	0.8157	0.8387
	CollegeMsg	0.5848	0.5665	0.5904	0.4787	0.6249	0.6253	0.5345	0.5635	0.6365
	MIT	0.7363	0.7324	0.7144	0.5401	0.3924	0.7442	0.6061	0.7648	0.7180
AUC	fb-forum	0.6293	0.6810	0.6294	0.5000	0.5912	0.7305	0.6215	0.6470	0.6552
	radoslaw-email	0.7047	0.6678	0.6079	0.5000	0.4101	0.6994	0.7694	0.7038	0.8197
	Eu-core	0.8109	0.8319	0.7957	0.5000	0.4167	0.8339	0.8348	0.8116	0.8369
	CollegeMsg	0.5851	0.5670	0.5909	0.5000	0.6231	0.6799	0.5464	0.5636	0.6421
	MIT	0.7441	0.7334	0.7125	0.5000	0.4139	0.7465	0.6210	0.7654	0.7284
PRECISION	fb-forum	0.6739	0.7182	0.6993	0.2953	0.6409	0.8888	0.7847	0.8243	0.8674
	radoslaw-email	0.7367	0.6841	0.6173	0.5304	0.2887	0.7280	0.7911	0.7459	0.8490
	Eu-core	0.8824	0.8960	0.8371	0.2405	0.3414	0.8807	0.9410	0.9081	0.9715
	CollegeMsg	0.7355	0.6648	0.7298	0.1447	0.6252	0.8263	0.7395	0.7969	0.8497
	MIT	0.8125	0.7636	0.7211	0.5401	0.3336	0.7892	0.7207	0.8152	0.8414
RECALL	fb-forum	0.4915	0.5124	0.4545	0.6000	0.4877	0.5310	0.3441	0.3769	0.3651
	radoslaw-email	0.6818	0.7076	0.6137	1.0000	0.0940	0.6892	0.7747	0.6751	0.8279
	Eu-core	0.7118	0.7534	0.7256	0.5000	0.2160	0.7699	0.7141	0.6890	0.6727
	CollegeMsg	0.2666	0.2756	0.2904	0.3000	0.6744	0.4498	0.1598	0.1732	0.5718
	MIT	0.6845	0.6859	0.7493	1.0000	0.1368	0.6957	0.4750	0.6931	0.5810
AUPR	fb-forum	0.7091	0.7725	0.7130	0.7510	0.6950	0.8300	0.7297	0.7584	0.7735
	radoslaw-email	0.7931	0.7737	0.7167	0.7652	0.4372	0.7909	0.8433	0.7981	0.8750
	Eu-core	0.8671	0.8870	0.8487	0.7499	0.4803	0.8822	0.8989	0.8738	0.9029
	CollegeMsg	0.6845	0.6519	0.6878	0.7553	0.7320	0.7723	0.6662	0.6916	0.7648
	MIT	0.8361	0.8041	0.8013	0.7700	0.4660	0.8222	0.7424	0.8317	0.8733
F1 MEASURE	fb-forum	0.6209	0.6811	0.6165	0.3282	0.5802	0.7154	0.5843	0.6161	0.6237
	radoslaw-email	0.7031	0.6678	0.6037	0.3465	0.3293	0.6969	0.7684	0.7010	0.8311
	Eu-core	0.8107	0.8300	0.7949	0.3248	0.3822	0.8135	0.8122	0.8112	0.8232
	CollegeMsg	0.5377	0.5253	0.5493	0.3237	0.6194	0.6644	0.4600	0.4835	0.6040
	MIT	0.7355	0.7309	0.7117	0.3503	0.3474	0.7424	0.6015	0.7631	0.7143

Here the proposed model *PQCLP-VQC* first gets 8 features from local and global categories from 4 snapshots, a total of 32 features, and the rest of the models are provided each feature from 4 snapshots i.e. 4 features.

moderate on fb-forum, and low on MIT. Now putting a light on the [Table 6](#) our model *PQCLP-VQC_LG* has the best AUC and F1 scores for radoslaw-email and eu-core. Next, like other models, it has a low recall on all the other datasets except radoslaw-email where it is the best performer. In terms of accuracy, it has low performance on fb-forum and moderate on MIT, and best on radoslaw-email, eu-core, and collegemsg. Similarly, for AUPR it has low performance on fb-forum, narrowly missing the best on collegemsg and best on radoslaw-email, eu-core, and MIT. However, it has the best performance in the case of precision for all the datasets except fb-forum where it narrowly misses the best score. Henceforth, it is evident that the proposed models perform better than the individual features.

Now, looking at the performance of the local feature model and global feature model for *QNN*, they both outperform the mixed feature model for *PQCLP-QNN_LG* which is provided both local and global features. Similar observations can be derived for *PQCLP-VQC_LG* variants. The justification for this is that when we combine the features and the number of features becomes large, the dimension needs to be reduced in order to address the qubit limitation for quantum models as stated earlier in [Section 4.1.1](#), the bigger is the feature set, the greater the data loss, also combining the features from mixed categories alter the performance of the overall model. In light of the fact that global features are typically more accurate and robust than local features when this feature is combined with local features performing poorly on these low-density datasets, they underperform naturally. But naturally, this is not evident in separate local and global comparisons where the mixing of categories does not occur. So effectively the model behaves in the way it should and performs well when exposed to the right set of features considering its current physical limitations of limited Qubits.

6.4. Performance comparison of *PQCLP: QNN* and *VQC* with their classical counterparts *NN* and *SVM* on local, global, and combined feature category

Having presented the comparison of different categories of models in the quantum domain we finally present here, the performance of

both the quantum models *PQCLP-QNN* and *PQCLP-VQC* with their classical counterparts i.e. *Neural-Networks* and *SVM* respectively. Here again, we present three categories of comparisons of *PQCLP* models i.e. local, global, and combined mode, where local variants both quantum and classical are provided with 4 local network metrics, global variants are provided with 4 global and quasi-local network metrics, and finally, combined models provided with 8 network metrics from all the three categories. The hyper-parameters of the quantum models namely *PQCLP-QNN_Local/PQCLP-VQC_Local*, *PQCLP-QNN_Global/PQCLP-VQC_Global*, *PQCLP-QNN_LG*, and *PQCLP-VQC_LG* are same as previously discussed models. For *NeuralNetwork* we use a learning rate of 0.001 and *BinaryCrossEntropy* loss with metrics like accuracy, AUC, Precision, and recall. For *SVM* we choose a linear kernel with a second-degree polynomial kernel function. Here we first present the comparison of *QNN* with its classical counterpart *NN(Neural Network)* and *VQC* with its classical counterpart *SVM(Support Vector Machine)* on the local domain, and global domain, and finally a comparison of features from both domains.

Observing the [Table 7](#) we may see that for local category our model *PQCLP-QNN_Local* outperforms the classical *NeuralNetwork* in Accuracy, AUC, and Precision for almost all the datasets. This model has a better recall score in just two datasets namely eu-core and MIT but lesser on fb-forum, collegemsg, and radoslaw-email. Whereas AUPR and F1 scores are good in eu-core, radoslaw-email, and MIT but lesser than *NN* on fb-forum and collegemsg. *PQCLP-VQC_Local* seems to perform better than *NN* on all the datasets for most of the metrics except for recall, as usual, performing low on fb-forum and collegemsg.

For global comparison we have a competitive performance from both classical and quantum models as follows: our global model *PQCLP-QNN_Global* has a lower performance on three datasets namely fb-forum, eu-core, and MIT for Accuracy, AUC, Recall, and AUPR. Performing worst in case of F1 being high only for collegemsg. But for precision, it is higher on all datasets except MIT. Now looking at

Table 7

Comparison and analysis of the proposed PQCLP:QNN with its classical counterpart NN and PQCLP:VQC with its classical counterpart SVM on a local set of features, global set of features, and finally set of combined features from both the category.

Metric	Dataset	Local features				Global features				Combined features			
		NN	QNN	SVM	VQC	NN	QNN	SVM	VQC	NN	QNN	SVM	VQC
Test Accuracy	fb-forum	0.5757	0.5772	0.5905	0.5674	0.8686	0.7335	0.8353	0.7724	0.6714	0.6806	0.7890	0.6579
	radoslaw-email	0.7559	0.7697	0.7694	0.8050	0.7705	0.7816	0.7767	0.8141	0.7875	0.7935	0.7822	0.8196
	Eu-core	0.6041	0.7678	0.6215	0.8351	0.9257	0.7891	0.9393	0.8250	0.8363	0.8091	0.8717	0.8387
	CollegeMsg	0.5355	0.5503	0.6081	0.5756	0.5915	0.6234	0.5330	0.6609	0.5817	0.6301	0.5289	0.6365
	MIT	0.6717	0.7677	0.6201	0.7411	0.7758	0.6397	0.7655	0.7798	0.7563	0.6797	0.8157	0.7180
AUC	fb-forum	0.7120	0.5606	0.5914	0.5691	0.9335	0.7300	0.8361	0.7660	0.8341	0.6722	0.7895	0.6552
	radoslaw-email	0.7650	0.8275	0.7607	0.8082	0.7718	0.8342	0.7838	0.8142	0.7718	0.8993	0.7865	0.8197
	Eu-core	0.7626	0.7679	0.6224	0.8349	0.9799	0.7912	0.9397	0.8226	0.9672	0.7912	0.9211	0.8369
	CollegeMsg	0.5485	0.5540	0.6077	0.5851	0.6168	0.6598	0.5317	0.6627	0.5918	0.6121	0.5293	0.6421
	MIT	0.7601	0.7718	0.6060	0.7441	0.8216	0.6442	0.7663	0.7802	0.8129	0.6442	0.8162	0.7284
PRECISION	fb-forum	0.5455	0.8257	0.6778	0.7746	0.8877	0.9559	0.8029	0.9744	0.7169	0.7875	0.7648	0.8674
	radoslaw-email	0.7587	0.8134	0.7373	0.8562	0.8122	0.8168	0.8382	0.8675	0.8077	0.8496	0.8323	0.8490
	Eu-core	0.7343	0.9502	0.5860	0.9625	0.9451	0.9657	0.9475	0.9749	0.9180	0.9603	0.9308	0.9715
	CollegeMsg	0.5503	0.7826	0.7549	0.8032	0.6682	0.9744	0.5328	0.9509	0.6749	0.7881	0.5149	0.8497
	MIT	0.6753	0.8077	0.5949	0.8056	0.7935	0.7217	0.7859	0.8573	0.7667	0.7218	0.8330	0.8414
RECALL	fb-forum	0.7841	0.1562	0.3613	0.1984	0.8567	0.4926	0.8851	0.5453	0.4587	0.6187	0.8383	0.4084
	radoslaw-email	0.8250	0.7689	0.7592	0.8039	0.7902	0.8443	0.6906	0.8160	0.7678	0.7337	0.8140	0.8279
	Eu-core	0.4016	0.5693	0.8624	0.6962	0.9015	0.8815	0.9306	0.6621	0.7653	0.7133	0.9052	0.6727
	CollegeMsg	0.8582	0.1495	0.2197	0.3768	0.4408	0.5670	0.5201	0.3439	0.2969	0.4256	0.3480	0.5718
	MIT	0.7562	0.7648	0.8633	0.6136	0.7948	0.7371	0.7599	0.6726	0.7696	0.4976	0.8087	0.5810
AUPR	fb-forum	0.7603	0.7063	0.6801	0.6877	0.9398	0.8490	0.8723	0.8702	0.8051	0.7642	0.8416	0.7735
	radoslaw-email	0.8390	0.8511	0.8417	0.8716	0.7381	0.8681	0.8773	0.8623	0.8443	0.8833	0.8745	0.8750
	Eu-core	0.7188	0.8692	0.7585	0.9051	0.9681	0.7810	0.9565	0.9017	0.9656	0.8815	0.9445	0.9029
	CollegeMsg	0.6794	0.6714	0.7091	0.7553	0.7147	0.7971	0.6495	0.8124	0.6847	0.6915	0.6485	0.7648
	MIT	0.7354	0.8465	0.7662	0.8139	0.8202	0.6182	0.8358	0.8470	0.7838	0.7371	0.8205	0.8733
F1 MEASURE	fb-forum	0.6477	0.2572	0.4690	0.4987	0.8700	0.6405	0.8415	0.7578	0.6314	0.6505	0.7985	0.6237
	radoslaw-email	0.7346	0.7800	0.8038	0.8292	0.7776	0.7337	0.8267	0.8361	0.8074	0.8128	0.8285	0.8311
	Eu-core	0.4853	0.7097	0.6851	0.8313	0.9244	0.6133	0.9387	0.8191	0.8682	0.7461	0.9193	0.8232
	CollegeMsg	0.5572	0.2461	0.4545	0.5143	0.6035	0.7100	0.3846	0.6228	0.4696	0.4951	0.5394	0.6040
	MIT	0.6972	0.7732	0.7025	0.7263	0.7883	0.4976	0.7718	0.7761	0.7643	0.5817	0.8201	0.7143

Note: In the Local Features category we compare the performance of PQCLP-QNN with its classical equivalent Neural Network (NN) similarly PQCLP-VQC with its classical equivalent Support Vector Machines (SVM) both provided a set of 4 local features from 4 snapshots a total of 16 features. Similarly for the Global Feature category PQCLP-QNN Vs. NN and PQCLP-VQC Vs SVM are provided 4 global features from 4 snapshots a total of 16 features. Finally PQCLP-QNN Vs. NN and PQCLP-VQC Vs SVM are provided 8 features from both the category and 4 snapshots a total of 32 features.

another global model PQCLP-VQC_Global, that has high Accuracy, AUC, and F1 measure on radoslaw-email, collegemsg, and MIT but low on 2 datasets i.e. fb-forum and eu-core. Worst in case of recall and highest in case of precision being best for all the datasets. The model has AUPR being higher on fb-forum, collegemsg, and MIT alongside giving a competitive score on other datasets.

From the combined category our models again give a competitive performance to their classical counterparts: PQCLP-QNN_LG has better Accuracy and F1 measure scores for fb-forum, radoslaw-email, and collegemsg and lower in the case of eu-core and MIT. It has lower AUC and AUPR scores for 3 datasets namely fb-forum, eu-core, and MIT, and higher for radoslaw-email and collegemsg. Next recall being lower on almost all datasets and precision being highest on all datasets, giving a balanced F1 score. PQCLP-VQC_LG has lower Accuracy, AUC, recall, and F1 score for 3 datasets namely fb-forum, eu-core, and MIT, and higher for radoslaw-email and collegemsg. The AUPR score for the model is high on radoslaw, collegemsg, and MIT and competitive on fb-forum and Eu-core. Note that the cause of high performing dataset like Eu-core to underperform is the limitation of qubit to compete for higher accuracy range achieved by classical models here. Precision is higher for all the datasets. Thus overall, PQCLP quantum models give a competitive performance in comparison to the classical models and outperform in the Local category.

To explore more the performance of these quantum models on different dataset sizes, we tried to change the test-set and train-set ratio to (10%, 20%, 30%, 40%, and 50%) and evaluated the performance in each case. Fig. 8, 9 show the effect of changing the test set ratio gradually on the performance of PQCLP-QNN_LG, PQCLP-VQC_LG, and their classical equivalents NN and SVM on different datasets based on Accuracy and Precision.

6.5. Analysis of accuracy pattern on different variations based on test-ratio

Looking at the graph in Fig. 8, we can observe that our PQCLP models VQC and QNN, having combined features from both local and global categories of network metrics perform better than classical models NN and SVM on datasets like CollegeMsg and radoslaw-email with genuine test set ratio i.e. 20% and beyond. We may also observe that collegemsg is a low-density dataset, having fewer connections per node, but still, the model is able to outperform classical models. In the other two datasets MIT and eu-core PQCLP, models give a narrow difference in genuine test set ratio as compared to NN and SVM. The only exception is the fb-form where the dataset is not only sparse but also small and insufficient for training in quantum space.

6.6. Analysis of precision pattern on different variations based on test-ratio

Looking at the graphs in Fig. 9, We may clearly observe that the PQCLP models viz QNN and VQC with all categories features outperform the classical models, the only exception is the MIT dataset where it narrowly misses the best performance that too only on lower test set ratio of 10% and 20%. This emphasizes the fact that the measurements by quantum models are very consistent giving more reliability to the predictions.

6.7. Performance comparison of proposed PQCLP with QNN and VQC with various state-of-the-art algorithms

We have compared our results with several state-of-art methods and presented a comparison here. Table 8 compares four different

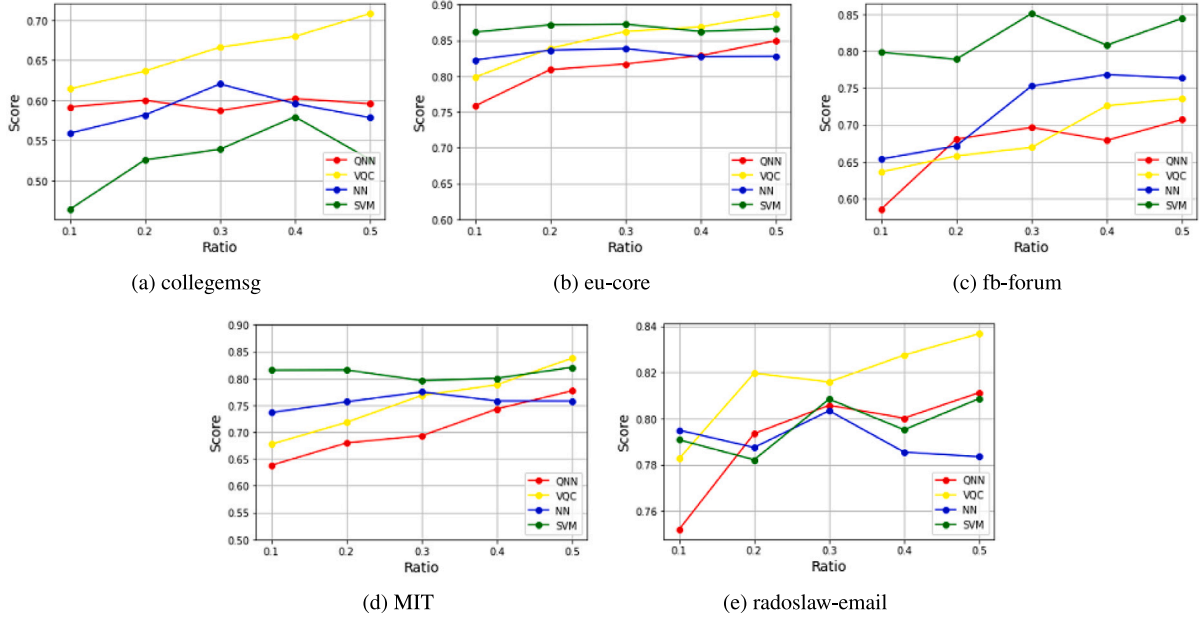


Fig. 8. Comparison of Accuracy score of Quantum models Vs. classical models for different test set ratios.

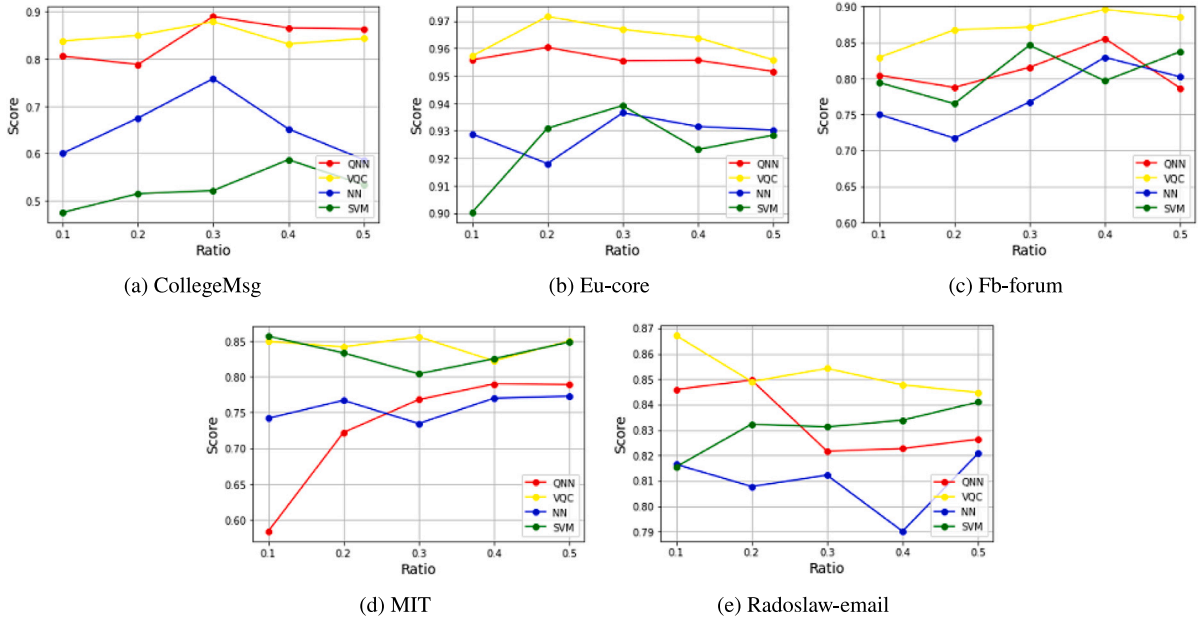


Fig. 9. Comparison of Precision score of Quantum models Vs. classical models for different test set ratios.

State-of-the-art methods with all of our quantum models viz. *PQCLP-QNN_Local/PQCLP-VQC_Local*, *PQCLP-QNN_Global/PQCLP-VQC_Global*, and *PQCLP-QNN_LG/PQCLP-VQC_LG* on five evaluation metrics. As we are comparing a number of different methods that have very competitive scores, we highlight here a few of the top scores that are closer to each other and differ by a smaller margin. Some of the popular state-of-the-art methods from the domain link prediction used in the comparison are:

- WEAK [88] — The authors of this paper have suggested the deployment of weak estimators in addition to the usage of standard similarity metrics in order to efficiently and cheaply generate a feature vector for a deep neural network. Weak estimators have been used in many machine learning techniques to improve model accuracy because of their capacity to forecast the shifting

probabilities in dynamic systems. A method for producing a computationally efficient, network-size independent feature vector is suggested in this study. For real-time applications and neural network link predictions, this feature vector is excellent. In this study, weak estimators are used with dynamic networks and similarity measures from static networks.

- CTDNE [89] — This paper provides a comprehensive framework for network embedding methods that take into account temporal data. The framework provides a mechanism for learning time-respecting embedding from continuous-time dynamic networks. Overall, the trials demonstrate the effectiveness of the proposed framework and dynamic network embedding technique. The results show that to develop acceptable and useful network representations, it is essential to characterize temporal dependencies in graphs.

Table 8Performance comparison and analysis of *PQCLP* models *VQC* and *QNN* with several state-of-the-art methods.

Metric	Dataset	WEAK	CTDNE	LGQ	DWALK-GNN	PQCLP-QNN			PQCLP-VQC		
						Local	Global	Combined	Local	Global	Combined
Accuracy	fb-forum	0.77917	0.44357	0.77333	0.8741	0.5772	0.7335	0.6806	0.5674	0.7724	0.6579
	radoslaw-email	0.76115	0.64278	0.78756	0.4887	0.7697	0.7816	0.7935	0.805	0.8141	0.8196
	Eu-core	0.79526	0.48953	0.80921	0.6088	0.7678	0.7891	0.8091	0.8351	0.825	0.8387
	CollegeMsg	0.61155	0.54379	0.60931	0.6613	0.5503	0.6234	0.6301	0.5756	0.6609	0.6365
	mit	0.75131	0.65414	0.76247	0.5549	0.7677	0.6397	0.6797	0.7411	0.7798	0.718
AUC	fb-forum	0.60916	0.67769	0.61677	0.8718	0.5606	0.73	0.6722	0.5691	0.766	0.6552
	radoslaw-email	0.62096	0.76239	0.69656	0.5102	0.8275	0.8342	0.8993	0.8082	0.8142	0.8197
	Eu-core	0.70698	0.66655	0.7041	0.4861	0.7679	0.7912	0.7912	0.8349	0.8226	0.8369
	CollegeMsg	0.56506	0.58023	0.4975	0.6716	0.554	0.6598	0.6121	0.5851	0.6627	0.6421
	mit	0.64211	0.73753	0.71667	0.5606	0.7718	0.6442	0.6442	0.7441	0.7802	0.7284
Precision	fb-forum	0.56243	0.20739	0.61639	0.7952	0.8257	0.9559	0.7875	0.7746	0.9744	0.8674
	radoslaw-email	0.39475	0.33307	0.66946	0.4646	0.8134	0.8168	0.8496	0.8562	0.8675	0.849
	Eu-core	0.56149	0.21793	0.6281	0.7732	0.9502	0.9657	0.9603	0.9625	0.9749	0.9715
	CollegeMsg	0.35622	0.20093	0.21995	0.5804	0.7826	0.9744	0.7881	0.8032	0.9509	0.8497
	mit	0.52029	0.36101	0.52329	0.5699	0.8077	0.7217	0.7218	0.8056	0.8573	0.8414
Recall	fb-forum	0.25349	0.8332	0.30029	0.8936	0.1562	0.4926	0.6187	0.1984	0.5453	0.3651
	radoslaw-email	0.28686	0.79523	0.35167	0.3671	0.7689	0.8443	0.7337	0.8039	0.816	0.8279
	Eu-core	0.50896	0.79937	0.49753	0.7318	0.5693	0.8815	0.7133	0.6962	0.6621	0.6727
	CollegeMsg	0.16978	0.55375	0.09488	0.7682	0.1495	0.567	0.4256	0.3768	0.3439	0.5718
	mit	0.43391	0.69537	0.49632	0.6149	0.7648	0.7371	0.4976	0.6136	0.6726	0.581
F1 score	fb-forum	0.29986	0.33186	0.3164	0.8422	0.2572	0.6405	0.6505	0.4987	0.7578	0.6237
	radoslaw-email	0.28884	0.46922	0.4208	0.3958	0.78	0.7337	0.8128	0.8292	0.8361	0.8311
	Eu-core	0.4384	0.3421	0.45022	0.7519	0.7097	0.6133	0.7461	0.8313	0.8191	0.8232
	CollegeMsg	0.27698	0.28913	0.21695	0.6613	0.2461	0.7101	0.4951	0.5143	0.6228	0.604
	mit	0.40064	0.47475	0.48289	0.5916	0.7732	0.4976	0.5817	0.7263	0.7761	0.7143

- LGQ [68] — A similarity-based link prediction approach that employs a similarity-based score is feature fusion-based link prediction in dynamic networks. The three most common similarity-based indices are Local (L), Global (G), and Quasi-local for determining similarity scores (Q). The broad categories of indices from the LGQ model are combined in a variety of ways (L, G, Q, LG, LQ, GQ, LGQ) to produce a feature set that may be used with different machine learning algorithms for link prediction. The Common Neighbors (CN), Adamic/Adar Index (AA), Jaccard Coefficient (JC), and Preferential Attachment (PA) local similarity indexes have all been considered (PA). In terms of global similarity indexes, cos+, Average Commute Time (ACT), Shortest Path (SP), and MFI (Matrix Forest Index) have also been taken into consideration. For quasi-local indices, the local path Index (LP) and Path of Length 3 have been taken into consideration (L3).
- DWALK-GNN [38]. Similar to how a convolutional neural network is created to operate on images, graph neural networks are an abstraction to multi-layer perceptrons that are tailored to operate on graphs rather than basic feature vectors. They utilize a convolutional filter to process the graph and extract information that is valuable locally. These latent representations can be used by statistical models to easily encode social relations in a continuous vector space. DeepWalk develops latent representations utilizing local information obtained from shortened random walks by treating walks as the equivalent of sentences. It recognizes structural patterns discovered during quick random walks. In this study, we use DeepWalk (DWALK) to learn latent vertices representations in a network to measure node distances, and then we use GNN to get the best link predictions that fit those distances. Concatenating DeepWalk embeddings from all of the results of the snapshot in the creation of more thorough features and labels. Then, using this information, a graph convolutional neural network is trained. This network includes graph convolutional layers from the Keras-dgl9 [109] library, dense and sigmoid activation layers from the TensorFlow Keras library, and more. Then, using this GNN, link prediction probabilities are predicted.

On careful examination of the Table 8, it can be easily visualized that all of our *PQCLP* models outperform the state-of-the-art methods

in most of the performance evaluation metrics for all the datasets except fb-forum, where the dataset is not only sparse but also small and insufficient for training in quantum space.

Starting with Accuracy, *PQCLP-QNN* performs moderately in all the variants as *PQCLP-QNN_LG* does well in the case of eu-core, radoslaw-email, and collegemsg, and *PQCLP-QNN_Local* well for MIT. The performance is low for fb-forum in all the variants. Now, for *PQCLP-VQC* the local variant *PQCLP-VQC_Local* has a good score for radoslaw-email and eu-core and a competitive score for MIT, global variant *PQCLP-VQC_Global* does well for all the datasets except fb-forum and the combined version *PQCLP-VQC_LG* does good for eu-core, radoslaw-email, and collegemsg and have an average score for MIT.

For AUC, *PQCLP-QNN* has a better score only for radoslaw-email, for the rest of the datasets, it stands below *PQCLP-VQC*. Whereas the *PQCLP-VQC_Global* variant has a better AUC score for all the datasets except fb-forum. *PQCLP-VQC_Local* performs well on eu-core, radoslaw-email, and MIT but low on fb-forum and collegemsg. *PQCLP-VQC_LG* has the best AUC score for eu-core, radoslaw-email and moderate AUC scores for collegemsg and MIT, and a low score for fb-forum.

For Precision, *PQCLP-QNN* does moderately well and *PQCLP-VQC* outperforms on all datasets. Looking at the details, *PQCLP-QNN_Local* has a better precision score on eu-core, radoslaw-email, and MIT and a moderate score on fb-forum and collegemsg. *PQCLP-QNN_Global* has the best scores on fb-forum, eu-core, and collegemsg and average scores on radoslaw-email and MIT. *PQCLP-QNN_LG* has better scores on radoslaw-email and eu-core but a competitive score on the rest of the datasets. *PQCLP-VQC_Global* performs best on all the datasets whereas *PQCLP-VQC_LG* does well on all the datasets except collegemsg where it has a moderate precision score. *PQCLP-VQC_Local* has a better precision score on radoslaw-email, eu-core, and MIT and a moderate score on fb-forum and collegemsg.

In case of a recall, as usual, our models suffer from low recall performance, *PQCLP-VQC* all the variants perform only in the case of radoslaw-email and poor on all other datasets. But to the surprise, *PQCLP-QNN_Global* performs best even better than CTDNE for all the datasets except fb-forum.

For the F1 score, *PQCLP-QNN_Local* has a good score only for MIT but low for all the other datasets. *PQCLP-QNN_Global* was able to

Table 9

Computational complexities of classical and quantum Machine Learning models. Here M is the dimension of the feature set, N is the number of training samples, W is the number of weights of the neural network model, δ is desired precision.

Classifier/Theoretical complexity	Classical	Quantum
Support vectors	$O(MXN)$	$O(\log(MXN))$
Neural network	$O(2^W)$	$O(4N^2 \log(N/\delta) 2^{W/2})$

perform only on fb-forum and collegemsg and *PQCLP-QNN_{LG}* is able to perform only on fb-forum and radoslaw-email while having a low score for all the other datasets. Now coming to the *PQCLP-VQC_{Local}*, it has a low F1 score for fb-forum and collegemsg but high on eu-core, radoslaw-email, and MIT. *PQCLP-VQC_{Global}* performs best on all the datasets whereas *PQCLP-VQC_{LG}* does well on all the datasets except fb-forum.

6.8. Efficiency comparison of proposed *PQCLP :QNN* and *VQC* with classical counterparts *NN* and *SVM*

Quantum machine learning is backed by the temporal speedup that we might accomplish with near-term quantum systems, according to the area's ongoing research. Recent works, such as those in [90], emphasize the use of quantum machine learning algorithms, particularly quantum *SVM* algorithms, which have markedly lower temporal complexity than their classical counterpart. The paper shows that the quantum variant of *SVM* can be implemented with the complexity of $O(\log(M \times N))$, which is a significant speedup over the classical variant of *SVM*. Liao et al. explain in the paper [91] that with reasonable weight parameters for neurons, we may have a semi-exponential to polynomial speed-up in quantum neural networks. This may be observed comparatively in the Table 9.

7. Conclusion

The goal of this work is to examine the consequences of completely employing quantum computing for machine learning, particularly for link prediction in dynamic networks. In order to accomplish this, we developed the *PQCLP* model, in which we took selected topological features from dynamic networks, projected them into quantum space, and thus benefited from the high-dimensional Hilbert space. We then used a parameterized quantum circuit to optimize the parameters once more in quantum space. Our suggested model employs two classification techniques, *PQCLP-QNN* and *PQCLP-VQC*, which have architectural similarities to neural networks and support vector machines, respectively. For the purpose of validating our feature set, we compare the overall performance of each category (local, global, local, and global together) to the individual features, i.e., the local feature-set (*PQCLP-QNN_L* and *PQCLP-VQC_L*), the global feature-set (*PQCLP-QNN_G* and *PQCLP-VQC_G*), and the feature-set from both the categories (*PQCLP-QNN_{LG}* and *PQCLP-VQC_{LG}*) on five well-known dynamic datasets over six different evaluation metrics. The outcomes support our model's ability to function well when given the appropriate feature combinations. The performance of our quantum models was then compared to that of their classical counterparts at various test-set ratios, and it was found that AUC exhibits increasing patterns as test data size increases and that Precision is generally high at test-set ratios of 20%–40%, with *PQCLP-VQC* outperforming it on the majority of datasets. We assessed the effectiveness of two of our models, *PQCLP-QNN* using neural networks and *PQCLP-VQC* with support vector machines and local, global, and mixed topological features. In comparison to conventional models, quantum models perform competitively. A comparison of all of our models to some base-lined state-of-the-art methods, such as WEAK, CTDNE, LGQ, and DWALK-GNN, demonstrates that all of our models outperform these earlier methods, with the global variation of *VQC* performing remarkably well for the majority of the evaluation criteria.

The goal of research on the link prediction job for dynamic time-varying networks is to increase accuracy and efficiency. Having better performance in terms of scores of various evaluation metrics and the speedup as elaborated in previous sections with quantum models, we may have two-fold benefits i.e. enhanced accuracy as well as efficiency. It is obvious that quantum machine learning would outperform classical models and give consistent predictions with improved accuracy and high precision, not deviating significantly due to the high dimensional probability. This promotes further incorporation of this technology for discovering missing and promising links in propitious fields like knowledge graphs, multiplex networks, and heterogeneous networks, and with techniques like node embedding for semantic link prediction and many more. When training the machine learning model, we may expand our approach to include a feature selection aspect that will rank the similarity scores and use only the most significant ones.

CRedit authorship contribution statement

Nisha Singh: Conceptualization, Data curation, Formal analysis, Methodology, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Mukesh Kumar:** Conceptualization, Methodology, Project administration, Supervision, Writing – review & editing. **Bhaskar Biswas:** Conceptualization, Project administration, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data has been attached

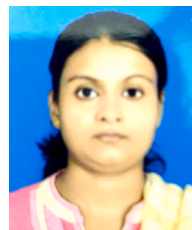
[Dynamic Dataset \(Reference data\)](#) (Mendeley Data.)

References

- [1] Z. Huang, D.D. Zeng, H. Chen, Analyzing consumer-product graphs: Empirical findings and applications in recommender systems, *Manage. Sci.* 53 (2007) 1146–1164, <http://dx.doi.org/10.2139/ssrn.906512>.
- [2] J. Chen, Y. Han, D. Li, J. Nie, Link prediction and route selection based on channel state detection in UASNs, *Int. J. Distrib. Sens. Netw.* 7 (1) (2011) <http://dx.doi.org/10.1155/2011/939864>.
- [3] A. Karaaslanli, S. Aviyente, Community detection in dynamic networks: equivalence between stochastic blockmodels and evolutionary spectral clustering, *IEEE Trans. Signal Inf. Process. Netw.* 7 (2021) 130–143, <http://dx.doi.org/10.1109/TSPN.2021.3052047>.
- [4] E.M. Airoldi, D.M. Blei, S.E. Fienberg, E.P. Xing, Mixed membership stochastic blockmodels, *J. Mach. Learn. Res.* 9 (65) (2008) 1981–2014, URL: <http://jmlr.org/papers/v9/airoldi08a.html>.
- [5] D. Koutra, U. Kang, J. Vreeken, C. Faloutsos, VoG: Summarizing and understanding large graphs, 2014, <http://dx.doi.org/10.48550/ARXIV.1406.3411>, URL: <https://arxiv.org/abs/1406.3411>.
- [6] N. De Cao, W. Aziz, I. Titov, Question answering by reasoning across documents with graph convolutional networks, 2018, <http://dx.doi.org/10.48550/ARXIV.1808.09920>, URL: <https://arxiv.org/abs/1808.09920>.
- [7] L. Lü, T. Zhou, Link prediction in complex networks: A survey, *Physica A* 390 (6) (2011) 1150–1170, <http://dx.doi.org/10.1016/j.physa.2010.11.027>, URL: <https://www.sciencedirect.com/science/article/pii/S037843711000991X>.
- [8] V. Martínez, F. Berzal, J.-C. Cubero, A survey of link prediction in complex networks, *ACM Computing Surveys* 49 (2016) 69, <http://dx.doi.org/10.1145/3012704>.
- [9] A. Kumar, S.S. Singh, K. Singh, B. Biswas, Link prediction techniques, applications, and performance: A survey, *Physica A* 553 (2020) 124289, <http://dx.doi.org/10.1016/j.physa.2020.124289>, URL: <https://www.sciencedirect.com/science/article/pii/S0378437120300856>.
- [10] C. Aggarwal, K. Subbian, Evolutionary network analysis: A survey, *ACM Comput. Surv.* 47 (2014) 10, <http://dx.doi.org/10.1145/2601412>.
- [11] H. Kashima, N. Abe, A parameterized probabilistic model of network evolution for supervised link prediction, in: Sixth International Conference on Data Mining, ICDM'06, 2006, pp. 340–349, <http://dx.doi.org/10.1109/ICDM.2006.8>.

- [12] S. Kerrache, R. Alharbi, H. Benhidour, A scalable similarity-popularity link prediction method, *Sci. Rep.* (2020) <http://dx.doi.org/10.1038/s41598-020-62636-1>.
- [13] M. Hasan, V. Chaoji, S. Salem, M. Zaki, Link prediction using supervised learning, 2006.
- [14] A. Zeguendry, Z. Jarir, M. Quafafou, Quantum machine learning: A review and case studies, *Entropy* 25 (2) (2023) <http://dx.doi.org/10.3390/e25020287>, URL: <https://www.mdpi.com/1099-4300/25/2/287>.
- [15] P. Rebentrost, M. Mohseni, S. Lloyd, Quantum support vector machine for big data classification, *Phys. Rev. Lett.* 113 (2014) 130503, <http://dx.doi.org/10.1103/PhysRevLett.113.130503>, URL: <https://link.aps.org/doi/10.1103/PhysRevLett.113.130503>.
- [16] E. Farhi, H. Neven, Classification with quantum neural networks on near term processors, *Quantum Phys.* (2018) URL: <https://api.semanticscholar.org/CorpusID:119037649>.
- [17] M.V. Altaisky, Quantum neural network, *Quantum Phys.* (2001) URL: <https://api.semanticscholar.org/CorpusID:14211220>.
- [18] A. Abbas, D. Sutter, C. Zoufal, A. Lucchi, A. Figalli, S. Woerner, The power of quantum neural networks, 2020, <http://dx.doi.org/10.1038/s43588-021-00084-1>.
- [19] I. Cong, S. Choi, M.D. Lukin, Quantum convolutional neural networks, *Nat. Phys.* (2019) <http://dx.doi.org/10.1038/s41567-019-0648-8>.
- [20] N. Wiebe, A. Kapoor, K.M. Svore, Quantum nearest-neighbor algorithms for machine learning, *Quantum Inf. Comput.* 15 (3–4) (2015) 318–358, URL: <https://www.microsoft.com/en-us/research/publication/quantum-nearest-neighbor-algorithms-for-machine-learning/>.
- [21] M. Schuld, I. Sinayskiy, F. Petruccione, Prediction by linear regression on a quantum computer, *Phys. Rev. A* 94 (2016) 022342, <http://dx.doi.org/10.1103/PhysRevA.94.022342>, URL: <https://link.aps.org/doi/10.1103/PhysRevA.94.022342>.
- [22] S. Lu, S. Braunstein, Quantum decision tree classifier, *Quantum Inf. Process.* 13 (2014) <http://dx.doi.org/10.1007/s11128-013-0687-5>.
- [23] I. Kerenidis, J. Landman, A. Luongo, A. Prakash, q-means: A quantum algorithm for unsupervised machine learning, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, Vol. 32, Curran Associates, Inc., 2019, URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/16026d60ff9b54410b3435b403af226-Paper.pdf.
- [24] M. Liu, J. Liu, R. Liu, H. Makhanov, D. Lykov, A. Apte, Y. Alexeev, Embedding learning in hybrid quantum-classical neural networks, in: 2022 IEEE International Conference on Quantum Computing and Engineering, QCE, IEEE, 2022, <http://dx.doi.org/10.1109/qce53715.2022.00026>.
- [25] M. Kumar, S. Mishra, B. Biswas, PQKLP: Projected quantum kernel based link prediction in dynamic networks, *Comput. Commun.* 196 (2022) 249–267, <http://dx.doi.org/10.1016/j.comcom.2022.10.006>, URL: <https://www.sciencedirect.com/science/article/pii/S0140366422003942>.
- [26] Z. Cao, Y. Zhang, J. Guan, S. Zhou, Link prediction based on quantum-inspired ant colony optimization, *Sci. Rep.* 8 (1) (2018) 13389, <http://dx.doi.org/10.1038/s41598-018-31254-3>.
- [27] J. Qian, L. Yang, Z. Yu, S. Liu, Link prediction using discrete-time quantum walk, *Teh. Vjesn.* 24 (2017) 1329–1334, <http://dx.doi.org/10.17559/TV-20160104014155>.
- [28] I. Djordjevic (Ed.), Copyright, in: *Quantum Information Processing and Quantum Error Correction*, Academic Press, Oxford, 2012, p. iv, <http://dx.doi.org/10.1016/B978-0-12-385491-9.02001-3>, URL: <https://www.sciencedirect.com/science/article/pii/B9780123854919020013>.
- [29] M. Schuld, N. Killoran, Quantum machine learning in feature Hilbert spaces, *Phys. Rev. Lett.* 122 (2019) 040504, <http://dx.doi.org/10.1103/PhysRevLett.122.040504>, URL: <https://link.aps.org/doi/10.1103/PhysRevLett.122.040504>.
- [30] V. Havlíček, A.D. Córcoles, K. Temme, A.W. Harrow, A. Kandala, J.M. Chow, J.M. Gambetta, Supervised learning with quantum-enhanced feature spaces, *Nature* 567 (7747) (2019) 209–212, <http://dx.doi.org/10.1038/s41586-019-0980-2>.
- [31] J. Preskill, Quantum computing in the NISQ era and beyond, *Quantum* 2 (2018) 79, <http://dx.doi.org/10.22331/q-2018-08-06-79>.
- [32] P. Reddy, A.B. Bhattacharjee, A hybrid quantum regression model for the prediction of molecular atomization energies, *Mach. Learn.: Sci. Technol.* 2 (2) (2021) 025019, <http://dx.doi.org/10.1088/2632-2153/abd486>.
- [33] J.P. Moutinho, A. Melo, B. Coutinho, I.A. Kovács, Y. Omar, Quantum link prediction in complex networks, *Phys. Rev. A* 107 (2023) 032605, <http://dx.doi.org/10.1103/PhysRevA.107.032605>, URL: <https://link.aps.org/doi/10.1103/PhysRevA.107.032605>.
- [34] M. Cerezo, A. Arrasmith, R. Babbush, S.C. Benjamin, S. Endo, K. Fujii, J.R. McClean, K. Mitarai, X. Yuan, L. Cincio, et al., Variational quantum algorithms, *Nat. Rev. Phys.* 3 (9) (2021) 625–644.
- [35] S. Boixo, S.V. Isakov, V.N. Smelyanskiy, R. Babbush, N. Ding, Z. Jiang, M.J. Bremner, J.M. Martinis, H. Neven, Characterizing quantum supremacy in near-term devices, *Nat. Phys.* 14 (6) (2018) 595–600.
- [36] F. Arute, K. Arya, R. Babbush, D. Bacon, J.C. Bardin, R. Barends, R. Biswas, S. Boixo, F.G. Brandao, D.A. Buell, et al., Quantum supremacy using a programmable superconducting processor, *Nature* 574 (7779) (2019) 505–510.
- [37] D. Liben-nowell, J. Kleinberg, The link prediction problem for social networks, *J. Am. Soc. Inf. Sci. Technol.* 58 (2003) <http://dx.doi.org/10.1002/asi.20591>.
- [38] A. Casteigts, *Finding structure in dynamic networks*, 2018.
- [39] M. Kumar, S. Mishra, B. Biswas, P.WAF: Path weight aggregation feature for link prediction in dynamic networks, *Comput. Commun.* 191 (2022) 438–458, <http://dx.doi.org/10.1016/j.comcom.2022.05.019>, URL: <https://www.sciencedirect.com/science/article/pii/S0140366422001712>.
- [40] M. Kumar, S. Mishra, R.D. Pandey, B. Biswas, CFLP: A new cost based feature for link prediction in dynamic networks, *J. Comput. Sci.* 62 (2022) 101726, <http://dx.doi.org/10.1016/j.jocs.2022.101726>, URL: <https://www.sciencedirect.com/science/article/pii/S1877550322001193>.
- [41] M. Newman, Newman, M.E.J.: Clustering and preferential attachment in growing networks. *Phys. Rev. E* 64, 025102, *Phys. Rev. E* 64 (2001) 025102, <http://dx.doi.org/10.1103/PhysRevE.64.025102>.
- [42] L.A. Adamic, N. Glance, The political blogosphere and the 2004 U.S. election: Divided they blog, in: *Proceedings of the 3rd International Workshop on Link Discovery, LinkKDD '05*, Association for Computing Machinery, New York, NY, USA, 2005, pp. 36–43, <http://dx.doi.org/10.1145/1134271.1134277>.
- [43] P. Jaccard, The distribution of the flora in the alpine zone. 1, *New Phytol.* 11 (2) (1912) 37–50.
- [44] A. Barabási, H. Jeong, Z. Neda, E. Ravasz, A. Schubert, T. Vicsek, Evolution of the social network of scientific collaborations, *Physica A* 311 (3) (2002) 590–614, [http://dx.doi.org/10.1016/S0378-4371\(02\)00736-7](http://dx.doi.org/10.1016/S0378-4371(02)00736-7), URL: <https://www.sciencedirect.com/science/article/pii/S0378437102007367>.
- [45] D. Liben-nowell, J. Kleinberg, The link prediction problem for social networks, *J. Am. Soc. Inf. Sci. Technol.* 58 (2003) <http://dx.doi.org/10.1002/asi.20591>.
- [46] K. Berahmand, E. Nasiri, S. Forouzandeh, Y. Li, A preference random walk algorithm for link prediction through mutual influence nodes in complex networks, *J. King Saud Univ. Comput. Inf. Sci.* 34 (8, Part A) (2022) 5375–5387, <http://dx.doi.org/10.1016/j.jksuci.2021.05.006>, URL: <https://www.sciencedirect.com/science/article/pii/S1319157821001099>.
- [47] W. Liu, L. Lü, Link prediction based on local random walk, *Europhys. Lett.* 89 (2010) 58007.
- [48] L. Lü, C.-H. Jin, T. Zhou, Similarity index based on local paths for link prediction of complex networks, *Phys. Rev. E* 80 (2009) 046122, <http://dx.doi.org/10.1103/PhysRevE.80.046122>.
- [49] Z. Wu, Y. Lin, J. Wang, S. Gregory, Link prediction with node clustering coefficient, *Physica A* 452 (2016) 1–8, <http://dx.doi.org/10.1016/j.physa.2016.01.038>, URL: <https://www.sciencedirect.com/science/article/pii/S0378437116000777>.
- [50] T. Zhou, Y.-L. Lee, G. Wang, Experimental analyses on 2-hop-based and 3-hop-based link prediction algorithms, *Physica A* 564 (2021) 125532, <http://dx.doi.org/10.1016/j.physa.2020.125532>.
- [51] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N.W.S. Lloyd, Quantum machine learning, *Nature* 549 (7671) (2017) 195–202, <http://dx.doi.org/10.1038/nature23474>.
- [52] M.A. Nielsen, I.L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2000.
- [53] N.H. Nguyen, *Quantum Neural Networks* (Ph.D. thesis), Wichita State University, College of Liberal Arts and Sciences, Dept. of Mathematics, Statistics and Physics, 2020.
- [54] A. Peres, D.R. Terno, Quantum information and relativity theory, *Rev. Modern Phys.* 76 (1) (2004) 93–123, <http://dx.doi.org/10.1103/revmodphys.76.93>.
- [55] Ł. Derkacz, M. Gwóźdź, L. Jakóbczyk, Entanglement beyond tensor product structure: algebraic aspects of quantum non-separability, *J. Phys. A* 45 (2) (2011) 025302, <http://dx.doi.org/10.1088/1751-8113/45/2/025302>.
- [56] T. Häner, D.S. Steiger, 0.5 petabyte simulation of a 45-qubit quantum circuit, in: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, ACM, 2017, <http://dx.doi.org/10.1145/3126908.3126947>.
- [57] R. de Wolf, *Quantum computing: Lecture notes*, 2019, <http://dx.doi.org/10.48550/ARXIV.1907.09415>.
- [58] I. Hamamura, T. Imamichi, Efficient evaluation of quantum observables using entangled measurements., *npj Quantum Inf.* 6 (2020) 56, <http://dx.doi.org/10.1038/s41534-020-0284-2>.
- [59] P. Sen, A.S. Bhatia, K.S. Bhangu, A. Elbeltagi, Variational quantum classifiers through the lens of the Hessian, in: T.R. Gadekallu (Ed.), *PLoS One* 17 (1) (2022) e0262346, <http://dx.doi.org/10.1371/journal.pone.0262346>.
- [60] H. Yano, Y. Suzuki, K. Itoh, R. Raymond, N. Yamamoto, Efficient discrete feature encoding for variational quantum classifier, *IEEE Trans. Quantum Eng.* 2 (2021) 1–14, <http://dx.doi.org/10.1109/tqe.2021.3103050>.
- [61] W. Li, D.-L. Deng, Recent advances for quantum classifiers, *Sci. China Phys. Mech. Astron.* 65 (2) (2021) <http://dx.doi.org/10.1007/s11433-021-1793-6>.
- [62] M. Schuld, A. Bocharov, K.M. Svore, N. Wiebe, Circuit-centric quantum classifiers, *Phys. Rev. A* 101 (3) (2020) <http://dx.doi.org/10.1103/physrev.101.032308>.
- [63] C.P. Gonçalves, Quantum neural machine learning - backpropagation and dynamics, 2016, <http://dx.doi.org/10.48550/ARXIV.1609.06935>, URL: <https://arxiv.org/abs/1609.06935>.
- [64] M. Rath, B. Pati, B. Pattanayak, An overview on social networking: Design, issues, emerging trends, and security, ISBN: 9780128154588, 2019, pp. 21–47, <http://dx.doi.org/10.1016/B978-0-12-815458-8.00002-5>.

- [65] V. Sekara, A. Stopczynski, S. Lehmann, Fundamental structures of dynamic social networks, *Proc. Natl. Acad. Sci.* 113 (36) (2016) 9977–9982, <http://dx.doi.org/10.1073/pnas.1602803113>, URL: <https://www.pnas.org/doi/abs/10.1073/pnas.1602803113>.
- [66] Y. Zou, R.V. Donner, N. Marwan, J.F. Donges, J. Kurths, Complex network approaches to nonlinear time series analysis, *Phys. Rep.* 787 (2019) 1–97, <http://dx.doi.org/10.1016/j.physrep.2018.10.005>, URL: <https://www.sciencedirect.com/science/article/pii/S03701571830276X>, Complex network approaches to nonlinear time series analysis.
- [67] H.-H. Jo, T. Hiraoka, Bursty time series analysis for temporal networks, in: P. Holme, J. Saramaki (Eds.), *Temporal Network Theory*, Springer International Publishing, Cham, 2019, pp. 161–179, http://dx.doi.org/10.1007/978-3-030-23495-9_9.
- [68] M. Kumar, S. Mishra, B. Biswas, Features fusion based link prediction in dynamic networks, *J. Comput. Sci.* 57 (2022) 101493, <http://dx.doi.org/10.1016/j.jocs.2021.101493>, URL: <https://www.sciencedirect.com/science/article/pii/S187750321001587>.
- [69] J. Spall, Adaptive stochastic approximation by the simultaneous perturbation method, *IEEE Trans. Automat. Control* 45 (10) (2000) 1839–1853, <http://dx.doi.org/10.1109/TAC.2000.880982>.
- [70] M.J.D. Powell, A direct search optimization method that models the objective and constraint functions by linear interpolation, in: S. Gomez, J.-P. Hennart (Eds.), *Advances in Optimization and Numerical Analysis*, Springer Netherlands, Dordrecht, 1994, pp. 51–67, http://dx.doi.org/10.1007/978-94-015-8330-5_4.
- [71] J. Liu, F. Tacchino, J.R. Glick, L. Jiang, A. Mezzacapo, Representation learning via quantum neural tangent kernels, *PRX Quantum* 3 (2022) 030323, <http://dx.doi.org/10.1103/PRXQuantum.3.030323>, URL: <https://link.aps.org/doi/10.1103/PRXQuantum.3.030323>.
- [72] J. Liu, K. Najafi, K. Sharma, F. Tacchino, L. Jiang, A. Mezzacapo, Analytic theory for the dynamics of wide quantum neural networks, *Phys. Rev. Lett.* 130 (2023) 150601, <http://dx.doi.org/10.1103/PhysRevLett.130.150601>.
- [73] R. Orús, Tensor networks for complex quantum systems, *Nat. Rev. Phys.* 1 (9) (2019) 538–550.
- [74] W. Huggins, P. Patil, B. Mitchell, K.B. Whaley, E.M. Stoudenmire, Towards quantum machine learning with tensor networks, *Quantum Sci. Technol.* 4 (2) (2019) 024001, <http://dx.doi.org/10.1088/2058-9565/aaea94>.
- [75] A. Klümper, A. Schadschneider, J. Zittartz, Matrix product ground states for one-dimensional spin-1 quantum antiferromagnets, *Europhys. Lett.* 24 (4) (1993) 293–297, <http://dx.doi.org/10.1209/0295-5075/24/4/010>.
- [76] Y.-Y. Shi, L.-M. Duan, G. Vidal, Classical simulation of quantum many-body systems with a tree tensor network, *Phys. Rev. A* 74 (2006) 022320, <http://dx.doi.org/10.1103/PhysRevA.74.022320>, URL: <https://link.aps.org/doi/10.1103/PhysRevA.74.022320>.
- [77] V. García, R.A. Mollineda, J.S. Sánchez, Index of balanced accuracy: A performance measure for skewed class distributions, in: H. Araujo, A.M. Mendonça, A.J. Pinho, M.I. Torres (Eds.), *Pattern Recognition and Image Analysis*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 441–448.
- [78] J. Davis, M. Goadrich, The relationship between precision-recall and ROC curves, in: *Proceedings of the 23rd International Conference on Machine Learning*, 2006, pp. 233–240.
- [79] T. Fawcett, An introduction to ROC analysis, *Pattern Recognit. Lett.* 27 (8) (2006) 861–874, <http://dx.doi.org/10.1016/j.patrec.2005.10.010>, URL: <https://www.sciencedirect.com/science/article/pii/S016786550500303X>, ROC Analysis in Pattern Recognition.
- [80] C. Goutte, E. Gaussier, A probabilistic interpretation of precision, recall and F-score, with implication for evaluation, in: D.E. Losada, J.M. Fernández-Luna (Eds.), *Advances in Information Retrieval*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 345–359.
- [81] D. Powers, Evaluation from precision, recall and F-factor to ROC, informedness, markedness and correlation, *Mach. Learn. Technol.* 2 (2008).
- [82] Y. Sasaki, The truth of the F-measure, *Teach. Tutor. Mater.* (2007).
- [83] J. Kunegis, Reality mining network dataset – KONECT, *J. Comput. Sci.* (2017) URL: <http://konect.cc/networks/mit>.
- [84] R. Michalski, S. Palus, P. Kazienko, Matching organizational structure and social network extracted from email communication, 87, 2011, pp. 197–206.
- [85] J. Leskovec, A. Krevl, SNAP Datasets: Stanford large network dataset collection, 2014, <http://snap.stanford.edu/data/email-Eu-core.html>.
- [86] R. Rossi, N. Ahmed, The network data repository with interactive graph analytics and visualization, in: AAAI, 2015, <http://networkrepository.com>.
- [87] J. Leskovec, A. Krevl, SNAP Datasets: Stanford large network dataset collection, 2014, <https://snap.stanford.edu/data/CollegeMsg.html>.
- [88] C. Chiu, J. Zhan, Deep learning for link prediction in dynamic networks using weak estimators, *IEEE Access* 6 (2018) 35937–35945, <http://dx.doi.org/10.1109/ACCESS.2018.2845876>.
- [89] G.H. Nguyen, J.B. Lee, R.A. Rossi, N.K. Ahmed, E. Koh, S. Kim, Continuous-time dynamic network embeddings, in: Companion Proceedings of the Web Conference 2018, in: WWW 18, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2018, pp. 969–976, <http://dx.doi.org/10.1145/3184558.3191526>.
- [90] Z. Li, X. Liu, N. Xu, J. Du, Experimental realization of a quantum support vector machine., *Phys. Rev. Lett.* 114 (2015) 140504.
- [91] Y. Liao, D. Ebler, F. Liu, O. Dahlsten, Quantum speed-up in global optimization of binary neural nets, *New J. Phys.* 23 (2021) <http://dx.doi.org/10.1088/1367-2630/abc9ef>.



Nisha Singh Received her Master of Science in Computer Science from Banaras Hindu University, Varanasi (Uttar Pradesh) in 2008 and Bachelor of Science in CS from Mithibai College, University of Mumbai, Mumbai (Maharashtra) in 2005. She is currently perusing her Ph.D. in Computer Science and Engineering and working as System Analyst cum Op. In-charge of Supercomputing Centre, Indian Institute of Technology (BHU), Varanasi. Her research interests include Quantum Machine Learning, Link Prediction, Community Detection, and Knowledge Graph completion.



Mukesh Kumar received Ph.D. in Computer Science and Engineering from Indian Institute of Technology (BHU), Varanasi and his Master of Engineering in Computer Networking from University Visvesvaraya College of Engineering (UVCE), Bangalore (Karnataka) in 2013. He received Bachelor of Engineering in Information Science & Engineering from Bapuji Institute of Engineering and Technology, Davangere (Karnataka) in 2007. He is currently working as associate professor in Kalinga Institute of Industrial Technology (KIIT), DU, Bhubaneswar. His research interests include Machine Learning, Quantum Computing, Link Prediction, Community Detection and Influence Maximization in social/complex networks.



Bhaskar Biswas received Ph.D. in Computer Science and Engineering from Indian Institute of Technology (BHU), Varanasi. He received the B.Tech. degree in Computer Science and Engineering from Birla Institute of Technology, Mesra. He is working as Associate Professor at Indian Institute of Technology (BHU), Varanasi in the Computer Science and Engineering department. His research interests include Data Mining, Text Analysis, Machine Learning, Influence Maximization, Link Prediction, Social Network Analysis.