

# Forward Selection for Time Series-Based Qubit Generation via Parameterized Quantum Gates

Singaraju Srinivasulu, Nagarajan G

Department of Computer Science and Engineering,

Sathyabama Institute of Science and Technology (Deemed to Be University), Chennai-119, Tamil Nadu, India

**Abstract**—Quantum data processing requires classical data to be encoded into quantum states. Current noisy intermediate-scale quantum devices have a limited number of qubits that are stable only briefly. Encoding classical data into qubits is the initial step in Quantum Machine Learning (QML), and effective encoding is crucial for quantum processing. This algorithms for data processing are still emerging, and compact data representations are essential for their success. This research proposes a novel data encoding technique using uniformly controlled rotation gates, achieving high storage density by encoding real-valued time series data as qubit rotations. The model uses a binary representation for computations on time series data, reducing the number of quantum measurements needed. The research explores quantum forward propagation in simulations to improve prediction accuracy for time series signals using parameterized quantum circuits, handling trends, noise, and sinusoidal components. The efficiency of the encoding process depends on data volume and chosen encoding, with potential infinite loading time in the worst case. This study presents a Forward Selection Time Series Data Processing and Feature Extraction Model for Qubits generation with Parameterized Quantum Gates (FSDPFEM-PQG), demonstrating superior performance in quantum representations compared to existing models.

**Keywords**—Quantum bits; Quantum Machine Learning; quantum algorithms; quantum measurements; Parameterized Quantum Gates; feature extraction; time-series data

## I. INTRODUCTION

The first generation of publicly traded quantum computers is the result of recent advancements in quantum technology. Some issues may be easier for quantum computers to tackle than for their classical counterparts [1]. A quantum computer may, in theory, do tasks like factoring big prime numbers or conducting an unstructured search more quickly [2]. These gains in speed are made possible by the fact that qubit-based quantum computers can take use of quantum mechanical phenomena like superposition and entanglement [3]. Due to their significant shortcomings, first-generation quantum computers have been dubbed Noisy Intermediate Scale Quantum (NISQ) devices [4]. Their qubits are unstable and noisy, decaying over a short period of time. The computers are considered to be medium-sized, with numbers of qubits ranging from dozens to hundreds. Hardware is projected to improve significantly, which will open the door to new uses for quantum computers. However, controlling these quantum devices is difficult because of the quantum nature of the devices and the hardware constraints that must be considered [5]. Quantum computers, in contrast to their conventional counterparts, handle data in a fundamentally different way. The qubits' states must be encoded with this information before they can be used. But

several data encodings dictate how information can be represented by qubits [6]. However, it is often assumed that loading data only takes logarithmic or linear time when an algorithm promises a linear or exponential speedup in theory. Given the limited number of operations that can be performed on noisy qubits, understanding these runtime consequences is especially important in the present NISQ era. Physicists and computer scientists need to work together to design software for quantum computers, and they need to have a common concept of data loading for this to be possible [7]. Quantum physics is proved to provide a more complete explanation of nature than classical physics. The computational complexity of simulating large-scale quantum systems using classical computers generally increases as the number of particles involved increases [8].

Qubits in quantum computers are realized using quantum particles, implying that, in principle, computational power may be enhanced exponentially by adding more controllable qubits to the system [9]. As a result, the fundamental benefit of quantum computers would be that they can circumvent classical computers' inability to simulate natural phenomena. Unfortunately, traditional data processing is still required for both state preparation and measurement in quantum computers due to the fact that input and output are still in ones and zeros [10]. As a result, the limits of complexity in classical computation are largely unaffected by the advent of quantum computing.

Machine Learning (ML) has become pervasive, with uses in fields as disparate as physics, medicine, and computer science as well as the more obvious image recognition, translation, and anomaly detection [11]. Although they are hindered by noise due to short coherence durations and restricted qubit connection, near-term quantum devices have received interest recently for their potential to address smaller examples of classical unsolvable issues [12]. In the recent decade, research into the intersection of quantum computing with machine learning, or Quantum Machine Learning (QML) has flourished [13]. There are a variety of subfields within QML itself that can be explored based on the classicality or quantumity of the data and the algorithm [14].

Quantum hardware has come a long way in the previous decade, with major improvements in areas such as qubit count, processing speed, and quantum volume [15]. As a result, there has been a rise in the quantity of works attempting to actualize QML on hardware. These efforts all have the same overarching goal: to show how useful quantum computers could be for real-world machine learning applications by using their special qualities like entanglement and superposition [16]. In order to fully grasp the potential and limitations of near-term quantum

devices in QML, an in-depth analysis is required [17].

In quantum computing, this geometric representation is significant because it illustrates how quantum gates—particularly single-qubit rotation gates—manipulate quantum states on the Bloch sphere and enable qubits to exist in superposition. Depending on the rotation axis, the angle  $\theta$  typically denotes a rotation applied by quantum gates such as  $R(\theta)$ ,  $R_y(\theta)$ , or  $R_z(\theta)$ . Since the introduction of noisy intermediate-scale quantum processors, QML has shifted its focus to a type of deep neural network. Most of these deep learning algorithms are implemented on Parameterized Quantum Circuits [18], a term which is now synonymous with QNNs [19]. The state of a readout can be related to the quantum encoding of a classical input by patterns that can be described by PQCs. PQCs have been found to achieve around 90% accuracy when applied to a sample of the MNIST database, which is comparable to the performance of a classical neural network with an equivalent number of parameters [20]. Learning a distribution of probabilities that can produce new samples identical to a given dataset is the challenge of generative modeling. Image and text production, anomaly detection [21], and data enrichment are just a few of the many uses for generative modeling. Interest in applying quantum computing to generative modeling has been on the rise as of late [22]. Quantum computing is a novel type of computing that represents data as quantum states and manipulates those states using quantum operations [23].

Some tasks related to machine learning may be exponentially faster on a quantum computer than on a classical computer. There has been substantial growth in the construction and programming of quantum computers in recent years, leading to the availability of multiple quantum computing platforms for study and experimentation. In this study, we advocate for the use of quantum gates in generative modeling. Using quantum circuits as both the generator and the discriminator, a quantum generative adversarial network approach is suggested. In an adversarial scenario, the proposed model learns complex probability distributions quickly with the help of a quantum version. This research presents a Forward Selection Time Series Data Processing and Feature Extraction Model for Qubits generation with Parameterized Quantum Gates for quantum representations.

### A. Problem Statement

The need to encode classical data into quantum states efficiently has been highlighted in strongly time-dependent, high-dimensional, and noisy time-series data by the rapid upsurge of QML. Conversely, Noisy Intermediate-Scale Quantum (NISQ) systems have few qubits, short coherence times and noisy gate interactions meaning they cannot utilize existing quantum data encoding and qubit generation methods. Compact data representation is also the primary concern of prior art such as Probabilistic Quantum Memory (PQM) and Quantum Random-Access Coding (QRAC), but lack scalability, redundancy in features, or the use of classical feature selection and extraction followed by quantum encoding. This means that quantum circuits get frequently used to encode the features which are irrelevant or partially informative, and the sensitivity to noise increases, and the quality of quantum representations declines. The accuracy of prediction is low and these models do not fit

complex time-series data as they lack ordered preprocessing and prioritization of features.

Furthermore, instead of viewing data loading, feature extraction and qubit creation as a pipeline, most of the current quantum encoding models deal with them as independent processes. Such a fragmented approach results in poor quality quantum state preparation and unnecessary overhead of processing because of poor mappings between parameterized quantum gates and classical features. To be able to meet the requirement of near-term quantum hardware, there is a need to balance representational expressiveness, minimal use of qubits, and low-circuit depth. This is no easy feat. A systematic and comprehensive framework is evidently needed to use parameterized quantum gates to encode meaningful representations to time-series data in order to preprocess it effectively; in particular, it is essential to select the most informative features, extract meaningful representations, and encode them into qubits. To enhance the scalability, accuracy, and resilience of quantum representations on the realistic limitations of NISQ machines, it is necessary to address this issue. To address these challenges, the present study will suggest a model of time-series data processing and feature extraction using forward-selection. Such strategy will enable better quantum representations and efficient production of qubits.

### B. Hypothesis

H1: Quantum representations on NISQ devices are highly more efficient and accurate using time-series data that has been filtered with forward feature selection before quantum encoding.

H2: Prior to the formation of qubits, the necessary and uninformatic removal of features simplifies the circuit, decreases the number of quantum gate operations in the circuit, and increases the circuit resistance to quantum noise and decoherence.

H3: The proposed method, with its built-in data processing, forward selection, feature extraction, and parametrized quantum gate-based qubit production pipeline, is more accurate in predictions than the conventional methods as well as the quantum-inspired baseline models.

### C. Research Contributions

The research contributions are:

- Reduced the feature redundancy and improved the utilization of the available limited quantum resources with a forward feature selection technique that is applicable in time-series data.
- With parameterized quantum gates, an algorithm is designed to efficiently generate qubits which encode some classical properties in succinct and expressive quantum codes.
- The proposed model can be easily adapted to NISQ devices since it has less computational overhead, less circuit depth and is optimized in qubit consumption.
- Its quantum gate processing accuracy, qubit generation efficiency, and the accuracy of feature selection were

demonstrated to have improved at any rate, and its accuracy in data processing, as well as in parameterizing quantum gates were shown to improve.

- Provided with the practical knowledge development towards the creation of robust, scalable quantum machine learning pipelines that can be used not only in time-series applications.

The proposed research study is organized as follows: Section I introduces quantum data encoding in noisy intermediate-scale quantum (NISQ) systems and outlines key scalability challenges. Section II reviews related work on quantum data representation and parameterized quantum circuits for time-series modeling. Section III presents the proposed uniformly controlled rotation-based encoding framework and quantum forward propagation mechanism for efficient time-series prediction. Section IV analyzes encoding efficiency, computational complexity, and scalability aspects, while Section V concludes with performance evaluation of the FSDPFEM-PQG model and future research directions.

## II. LITERATURE REVIEW

### A. Related Works

There has been a recent surge in interest in using quantum-improved methods to address various machine learning challenges. For supervised learning, a common approach is to employ a variational method, which combines classical computation with the quantum resources of imperfect quantum devices. The key to its success is the translation of classical traits into a quantum-improved one. While several quantum-mapping functions have been proposed, few have focused on the efficient mapping of discrete features that are typically crucial for identifying datasets of interest. To efficiently map such discrete features into a restricted number of qubits for variational quantum classification (VQC), [1] introduced the usage of quantum random-access coding (QRAC). The author showcased several encoding methods and showed their strengths and weaknesses using numerical simulations. The author demonstrated experimentally that QRAC's ability to save on the number of qubits required for the mapping can speed up the training of VQC.

By switching from one-hot encoding to label encoding, [2] expressed multi-attribute data more compactly; both encodings provide the same Hamming distance. This replacement can be easily implemented on a traditional computer. However, due to the bit-level nature of PQM and P-PQM, switching them out on a QC is not simple. The author introduced an improved version of P-PQM, which is named as efficient P-PQM (EP-PQM) since it decreases the circuit depth of the data storage and retrieval processes by allowing label encoding of data stored in a PQM data structure. The author proposed quantum encoding structures based on the proposed encoding techniques using one and two qudit gates, which is a further step toward the implementation of coded quantum systems for communication and computing using qudits.

The quick progress of image sensors allows for the production of high-resolution images. Due to computational limitations, many image processing algorithms necessitate thumbnail-sized images. As a result, many uses require that

the photos be preprocessed to decrease their size. In [3], the authors suggested a new method for image compression that makes use of quantum computing. In this research, the author analyzed and compare several popular approaches to quantum-safe data encoding. In order to reduce the file size of images, the author offered four quantum compression methods by elaborating on the unitary operations of amplitude embedding. The proposed techniques scale exponentially with the number of qubits used since  $2n$  classical data values can be encoded into  $n$  qubits via amplitude embedding. The proposed compression strategies are evaluated based on their compression performance, as well as their visual evaluation and objective evaluation. From the tests, at 75% compression, the important patterns in images are still visible.

The minimal-depth approach achieves a T -depth of  $O(\log(N))$ , while the minimal-count approach obtains a T -count of  $O(N\log(\log(N)))$ . In [5], the authors gave a modular circuit-level implementation and resource estimates for each of these approaches to block-encoding a dense NN matrix of classical data to precision. The author looked at the costs and benefits of each method, and investigates how two distinct quantum random access memory models can be implemented. As part of this study, the author presented a novel state preparation technique with T -depth  $O(\log(N))$  that scales better than prior constructions with  $O(\log^2(N))$  complexity. These findings extend beyond the realm of basic query complexity to shed light on the resource costs incurred by quantum algorithms when substantial volumes of classical data are expected to be at their disposal.

As the quantum age begins, several vital fields of research will get a leg up into the next century. For these theories to work, we need a physical mechanism that can store, analyze, and transmit quantum data for a significant amount of time. Current quantum computers are small and affected by many non-ideal phenomena; so, optimal engineering of these computers requires a successful approach for exploring the design space. So, it's critical to build hardware-aware simulation infrastructures that can mimic quantum hardware's actions by starting from the ground up and systematically looking for new and improved methods to do things. In [6], the authors delves into the process of creating a compact model-based classical simulation infrastructure for quantum computation with semiconductor quantum dots, where each device is described in terms of the main physical parameters affecting its performance in a computationally simple way that still yields accurate results without the need for complex physical simulators.

In this study, the NP-hard wireless channel assignment problem is recast as a Higher-order Unconstrained Binary Optimization (HUBO), and a quadratic speedup is procured by use of the Grover adaptive search (GAS). Using a one-hot encoding of the channel indices, the quadratic formulation is the basis of the classical method. The novel quantum circuit that [7] designed was based on envisioning both upward and downward binary encodings of the channel indices, which allowed them to determine the precise qubit and gate counts needed for GAS. This study demonstrates that the proposed HUBO formulation significantly decreases the quantity of qubits and the difficulty of queries in comparison to the more prevalent quadratic formulation. This advantage is achieved

with a reduced number of quantum gates using the proposed descending binary encoding.

The literature on the topic of classical-quantum interface currently focuses primarily on discrete systems, e.g. compact feature encoding, qubit reduction or circuit optimization, although quantum data encoding and quantum machine learning have made significant progress. Methods like Quantum Random-Access Coding (QRAC) seek to optimize the number of qubits required to map features, but Probabilistic Quantum Memory (PQM) and other types accept such systems seek to optimize data storage and retrieval using parameterized quantum systems. The extraction of features, feature selection and organized classical preprocessing are key processes prior to quantum encoding but they overlook the importance of these processes when quantum efficiency and encoding compactness are considered. Classification and generative modeling with parameterized quantum circuits (PQCs) have been studied to some degree, but most of those studies have assumed that the input properties are ideal or otherwise favourable to quantum encoding. It leads to more circuit depth, an increase in the number of gates being operated than is required, and a reduction in resilience to NISQ constraints. More so, time-series data feature selection algorithms are needed before qubit creation to support time-related correlations and high dimensionality, but such strategies are barely considered in the literature.

To address these limitations, this study introduces the FSDPFEM-PQG, a model to generate quantum gate qubits parameterized by forward selection time series data processing and feature extraction. The proposed paradigm, unlike in prior studies, uses data preprocessing, forward feature selection, feature extraction and parameterized quantum gate-based encoding. Combining these techniques, we will be able to save qubit redundancy, minimize circuit depth, and can increase the quality of quantum representations, by only mapping the most informative time-series features to quantum states. The current work is an addition to the literature that allows creating a relationship between classical feature engineering and useful quantum representation learning on the devices of the NISQ age. It achieves this by making the proposed model clearly visible in the context of the existing methods of encoding and PQC.

### III. PROPOSED MODEL

Quantum bits can store data gleaned from quantum sensing and quantum control operations. Qubits allow for efficient encoding of classical data, while the converse is not true. While classical information must be encoded into a quantum system, quantum data in QML is information that is already in a quantum state. The proposed architecture is designed as a structured hybrid classical-quantum pipeline that transforms raw time-series data into compact and expressive quantum representations. The architecture consists of sequential processing stages that ensure efficient utilization of limited quantum resources while maintaining high representational accuracy.

The architecture of the proposed model begins with Input Time Series Data layer in which the raw time-based data is collected with benchmark data. The information then is passed to Data Processing Module where the normalization is made, any noisy data is removed, any missing values and scaling

are done to prepare the data to be analyzed. It is a stabilizing effect that ensures the uniformity of the additional quantum encoding. The processed data is then inputted to the Forward Feature Selection Module which repeatedly picks the most informative features by one of a set of predefined threshold criteria. This step removes redundant and weakly correlated features to dimensionality in the architecture to avoid unnecessary qubits and is not preoccupied with allocating the qubits. It is a significant move towards NISQ action. The features that have been chosen are then fed to the Feature Extraction Module where interesting patterns and reduced dimensional representations will be acquired. The module ensures that significant time characteristics of the time-series data are preserved without excessive complexity of the computing. The obtained features are then fed to the Parameterized Quantum Gates Processing Module where the classical feature values are represented in the form of rotation parameters of quantum gates such as  $RX(\theta)$ ,  $RY(\theta)$ , and  $RZ(\theta)$ . CNOT entangling gates are used to describe the inter-feature correlations of the quantum circuit. The Qubit Generation Module, which converts the coded quantum parameters into qubit states, follows the gate processing, it is the literal translation of classical information to the quantum domain. Finally, the Quantum Representation Generation Module is a model that produces structured quantum representations which can be used in the downstream quantum machine learning task(s) of classification, regression or generative modelling. The proposed model framework is depicted in Fig. 1.

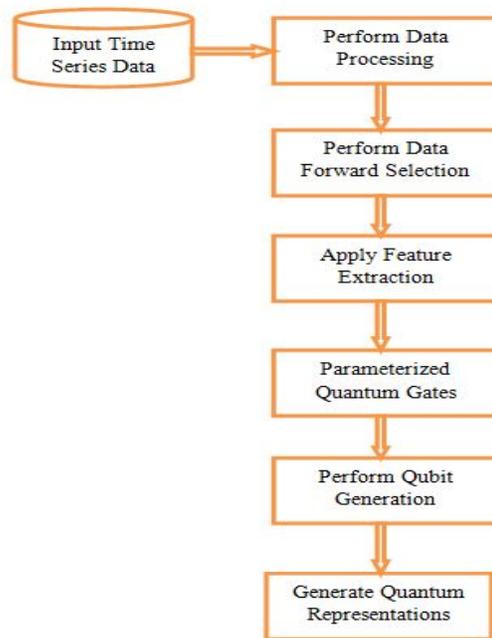


Fig. 1. Proposed model framework.

In order to prepare input time-series data for quantum machine learning or quantum-enhanced data processing, the proposed architecture depicts a sequential pipeline that does just that. The process starts with the input time-series data, which goes through a data processing stage to make sure it's in a usable format by cleaning, normalizing, and maybe

restructuring it. The next step is to use Data Forward Selection to improve model efficiency and decrease computational cost by identifying and retaining the most useful and relevant features from the dataset. For successful learning or classification tasks, it is essential to extract the underlying patterns or characteristics of the data, which is achieved by Feature Extraction. This method is applied to the selected features. Crucial in quantum computing, Parameterized Quantum Gates are configured using the transformed data.

These gates manipulate qubits in a controlled manner using input parameters obtained from the features. After that comes Qubit Generation, which functions as a translation from the classical to the quantum realm by encoding classical data into quantum bits. The last step is to generate quantum representations using these qubits; these will then serve as the foundation for subsequent quantum algorithms. The integration of classical and quantum time series data is made possible by this entire flow, which opens the door to the possibility of using quantum processing speed or model accuracy advantages.

It's not easy to encode information in qubits. The qubits in today's gadgets are only stable for short periods of time and there aren't very many of them. The representation needs to be small, employing just a few qubits and a small number of quantum gates, so that it may be used with existing hardware. Since qubits degrade quickly and quantum gates might introduce errors, only a minimal amount of operations are allowed to set up the quantum state. A logarithmic or linear runtime, i.e., the number of parallel operations in the state preparation process, is optimal for effectively encoding even a high number of data values. There is a tradeoff between the amount of qubits needed and the runtime complexity of the loading operation for each encoding. Moreover, an additional factor is that data be represented in a way conducive to subsequent processes. When performing arithmetic operations such as addition or multiplication, it is often necessary to represent the data as its exact value. Relative values may be all that's needed to indicate some operations. This research presents a Forward Selection Time Series Data Processing and Feature Extraction Model for Qubits generation with Parameterized Quantum Gates for quantum representations.

The parameterized quantum circuit (PQC) used in the illustrated pipeline typically takes the form of a variational quantum circuit composed of a sequence of parameterized quantum gates. These gates are applied to qubits to encode the extracted features from classical time series data into quantum states. Commonly, the PQC includes rotational quantum gates such as  $R(\theta)$ ,  $R_Y(\theta)$ , and  $R_Z(\theta)$ , where the rotation angles  $\theta$  are derived from the processed and selected features. These gates enable the transformation of classical information into quantum representations by rotating qubit states on the Bloch sphere.

Additionally, the circuit may incorporate entangling gates such as the CNOT gate to capture correlations between qubits, thereby enhancing the representational capacity of the quantum model. The symbol “|” seen in quantum formulas, such as in the ket notation  $|0\rangle$  or  $|\psi\rangle$ , represents a quantum state vector and is a standard component of Dirac notation in quantum mechanics. It does not denote integer division or conditional probability; rather, it indicates a qubit state within a

complex vector space, which forms the foundation of quantum computation.

Step-1: Data processing, or the transformation of raw data into actionable information, is typically carried out in stages by a team of data researchers and data engineers within an organization. This transformation may involve manual manipulation and analysis of the data, or it may be automated with the help of various features of an information management application. The pre processed data is further used for processing in Qubit generation. The pre-processing is performed as:

$$T_{\text{data}}[M] = \sum_{i=1}^M \left( \text{getattrVal}(\text{TSset}(i)) + \gamma(\text{TSset}(i)) \right) + \sum_{i=1}^M \text{getattrrange}(i) + \max_{0 \leq x \leq N} (xe^{-x^2}) + \frac{\gamma(\text{getattrranger}(\text{TSset}(i)))}{\text{len}(\text{TSset})} \quad (1)$$

Here,  $xe$  is the data units,  $\gamma$  is the model for getting maximum range of values and  $\delta$  is the model for handling missing values that will be replaced using the average of the attributes.  $M$  is the total records from the time-series data.

Step-2: The selection of predictive variables in stepwise regression is handled by an automated process, making it a popular method for fitting models of regression in statistics. Each stage involves evaluating whether a given variable should be added to or removed from the pool of explanatory variables. Forward selection begins with no variables in the model, then tests the addition of all of them using a chosen satisfactory model fit criterion, then adds the element to which it gives the most substantial enhancement of the fit, and so on until no additional variables improve the fit. The data forward selection process is performed as:

$$T_{\text{data\_norm}}[M] = \sum_{i=1}^M \left[ \frac{\text{getattrVal}(T_{\text{data}}(i)) - \text{MinVal}(i)}{\text{MaxVal}(i) - \text{MinVal}(i)} \times (\text{newMaxVal}(i) - \text{newMinVal}(i)) + \text{newMinVal}(i) \right] + \sum_{i=1}^M \left( \frac{\sum_{i=1}^N \text{getattrVal}(i)}{\delta(i)} \right) \quad (2)$$

Th is the threshold value for selecting the features. New low range values are provided to high range values.

Step-3: In feature extraction, the original, unstructured data is broken down into more digestible chunks, a process known as dimensionality reduction. A hallmark of these enormous data sets is a vast number of variables that demand a lot of computational resources to process. To reduce the amount of

---

**Algorithm 1** Algotihm-FSDPFEM-PQG : Quantum Feature Representation from Time-Series

---

**Require:** Time Series Dataset TSset

**Ensure:** Quantum Feature Representation QFRset

**Step 1: Data Pre-processing**

Transform raw time series data into processed form

$$T_{\text{data}}[M] = \sum_{i=1}^M \left( \text{getattrVal}(\text{TSset}(i)) + \gamma(\text{TSset}(i)) \right) \\ + \sum_{i=1}^M \text{getattrrange}(i) + \max_{0 \leq x \leq N} (xe^{-x^2}) \\ + \frac{\gamma(\text{getattrranger}(\text{TSset}(i)))}{\text{len}(\text{TSset})}$$

**Step 2: Data Normalization and Forward Selection**

$$T_{\text{data\_norm}}[M] = \sum_{i=1}^M \left[ \frac{\text{getattrVal}(T_{\text{data}}(i)) - \text{MinVal}(i)}{\text{MaxVal}(i) - \text{MinVal}(i)} \cdot (\text{newMaxVal}(i) - \text{newMinVal}(i)) \right. \\ \left. + \text{newMinVal}(i) \right] + \sum_{i=1}^M \left( \frac{\sum_{i=1}^N \text{getattrVal}(i)}{\delta(i)} \right)$$

**Step 3: Feature Selection**

**for**  $i = 1$  to  $M$  **do**

**if**  $\max(F_{\text{Select}}(i)) < Th$  **then**

$F_{\text{set}}(i) \leftarrow F_{\text{Select}}(i)$

**else**

    continue

**end if**

**end for**

**Step 4: Quantum Feature Encoding**

$$F_{\text{extrset}}[M] = \sum_{i=1}^M \left[ \frac{\text{getattr}(\text{MaxVal}(F_{\text{Select}}(i, i + 1)))}{\text{len}(F_{\text{Select}})} \right. \\ \left. + \left( \frac{\text{getMaxrange}(i, i + 1) + \text{newMaxVal}(i)}{M} \right)^2 \right. \\ \left. + \frac{\text{minRange}(F_{\text{set}}(i, i + 1))}{M} \right. \\ \left. + \sqrt{\frac{\text{getattrVal}(F_{\text{Select}}(i, i + 1)) + \min(\delta(i, i + 1))}{\text{len}(\text{TSset})}} \right]$$

**Step 5: Qubit Generation**

$$\omega[M] = \prod_{i=1}^B (\text{Data}(i) | F_{\text{extrset}}(i))$$

$$P_{\text{quantum}}[M] = \prod_{i=1}^{M(R)} \left( R_i + \max\text{attVal}(\omega(i)) + \cos\left(\frac{R_i}{2}\right) \right. \\ \left. | F_{\text{extrset}}(i) + \sin\left(\frac{R_{i+1}}{2}\right) | \text{Data}(i) \right)$$

**Step 6: Quantum Representation Generation**

$$\tau[M] = \sum_{i=1}^M \left( \max(P_{\text{quantum}}(i, i + 1)) + \max(\omega[M]) \right. \\ \left. + \cos\left(\frac{R_i}{2}\right) | \sin\left(\frac{R_i}{2}\right) \right)$$

QFRset  $\leftarrow \tau[M]$

data that needs to be processed while still providing an accurate and complete description of the original time series data set, the feature extraction process is performed as:

$$F_{\text{Select}}[M] = \begin{cases} F_{\text{set}}(i) \leftarrow F_{\text{Select}}(i), & \text{if } \max(F_{\text{Select}}(i)) < Th, \\ \text{continue}, & \text{otherwise.} \end{cases} \quad (3)$$

Step-4: Quantum algorithms that can vary in response to changes in input parameters are called variational or parameterized quantum circuits. Unitary matrices are used to depict quantum logic gates. Standard quantum gates typically function on one or two qubit spaces. Matrix descriptions of quantum gates can be written in terms of 4x4 matrices with orthonormal rows. The parameterized quantum gates are processed. The data Di with N dimensional vector, is encoded to quantum state. The encoding of quantum state of the feature set is:

$$F_{\text{extrset}}[M] = \sum_{i=1}^M \left[ \frac{\text{getattr}(\text{MaxVal}(F_{\text{Select}}(i, i+1)))}{\text{len}(F_{\text{Select}})} \right. \\ \left. + \lim_{i \rightarrow M} \left( \frac{\text{getMaxrange}(i, i+1) + \text{newMaxVal}(i)}{M} \right)^2 \right. \\ \left. + \frac{\text{minRange}(F_{\text{set}}(i, i+1))}{M} \right. \\ \left. + \sqrt{\frac{\text{getattrVal}(F_{\text{Select}}(i, i+1)) + \min(\delta(i, i+1))}{\text{len}(\text{TSset})}} \right] \quad (4)$$

A quantum circuit required a minimum  $O(\log M)$  depth circuits from the records R, where each attribute i that belongs to R is the parameter of normalized quantum bits. The parameterized quantum gates are processed as given below.

Here,  $\omega$  represents the encoded quantum bits.

Step-5: The bit is the fundamental unit of information in conventional computing, and its quantum counterpart is a qubit. While both qubits and bits serve to store information, qubits are based on quantum characteristics and hence behave very differently. The qubit generation is performed as given below.

A qubit is represented as a data point of a feature vector that is indicated as:

$$\omega[M] = \prod_{i=1}^B (\text{Data}(i) \mid F_{\text{extrset}}(i)) \quad (5)$$

The qubit cartesian product is calculated as:

$$P_{\text{quantum}}[M] = \prod_{i=1}^{M(R)} \left( R_i + \text{maxattVal}(\omega(i)) + \cos\left(\frac{R_i}{2}\right) \right. \\ \left. \mid F_{\text{extrset}}(i) + \sin\left(\frac{R_{i+1}}{2}\right) \mid \text{Data}(i) \right) \quad (6)$$

Step-6: This work presents a novel quantum model for time-series data, one that is sensitive to the length of the intervals being studied. This method can be applied to the

discretization and modeling of logical connections. Then, to maximize performance, these logical connections are modified in a number of ways and quantum representations are generated as:

$$\tau[M] = \sum_{i=1}^M \left( \max(P_{\text{quantum}}(i, i+1)) + \max(\omega[M]) \right. \\ \left. + \cos\left(\frac{R_i}{2}\right) \mid \sin\left(\frac{R_i}{2}\right) \right) \quad (7)$$

Here,  $\xi$  represents the qubit cartesian product.

where, T is the quantum bits represented to generate a quantum matrix representation (see Algorithm 1).

The proposed FSDPFEM-PQG model offers a methodological framework and a unified system of transforming the classical time-series data into efficient quantum models. In comparison to the modern approaches of quantum encoding, where quantum encoding and classical preprocessing are treated as separate encoding stages, the above-presented model has the benefit of enforcing that only the most informative and most compact features are encoded in qubits by quantum encoding. This significantly reduces the depth of circuits, constrains the operations that can be performed on gates and increases resistance to quantum noise as well as quantum decoherence. The parameterized quantum gate functionality is an addition that provides further expressiveness to the quantum states generated and does not bother near-term quantum hardware. All in all, the proposed architecture is a robust foundation of the efficient qubit creation, and quality quantum representations with regard to time-series data. The efficiency of the model is proved in the following Section IV with a detailed analysis of the experiments and the comparison with the state-of-the-art procedures of quantum encoding.

## IV. RESULTS AND DISCUSSION

### A. Simulation Procedure

The bit has served as computers primary data-encoding unit for many years. The first commercial quantum computers, which use qubits rather than bits, have been developed in recent years. A qubit, like a bit, can be in one of two states: 0 or 1, representing the values 0 and 1. But quantum mechanics also allows it to be in a superposition of states, in which both of these possibilities coexist. Due to its use of superposition and other quantum mechanical features, quantum computers may be able to do calculations far more quickly than traditional computers. Several algorithms, such as the factorization of prime numbers, have been developed for quantum computers in which potential linear or exponential speedups over their conventional equivalents have been demonstrated. More and more businesses are investigating quantum computing as the number of available qubits on quantum computers grows. It is predicted, however, that devices in the foreseeable future will only have a few hundred qubits. The imperfect nature of these qubits is also a limitation. Their steady states don't last long at all. There is a finite number of actions that can be performed on them before they decay completely. Therefore, the current hardware constraints prevent us from successfully programming quantum computers.

In addition, from the perspective of a program designer, strange things happen because quantum computers must follow the laws of quantum mechanics. Quantum state access is the initial step in reading a qubit. Only through measurement can this be possible. Each algorithm processes data by conducting computations based on the assumption that a certain encoding scheme was used. There are times when data loading is inefficient. In the worst-case scenario, loading takes an infinite amount of time. Algorithms that would normally have a logarithmic or linear runtime are slowed down by this: Their overall execution time is similarly exponential because of the exponential loading time. Some data encodings and algorithms necessitate this procedure. Information is then fed into a quantum computer, as follows: When all qubits are set to  $|0\rangle$  at the outset, the resulting quantum state is written as  $|00\dots 0\rangle$ . In order to alter the state of a register of qubits, a state preparation routine must be applied to it. As a result, a quantum state need to be considered that codes for the information in question. The patterns associated with quantum states can characterize some aspects of this state. The quantum register is then used for calculations after the state has been prepared.

These calculations are represented as quantum gates, and they are unitary transformations. This research presents a Forward Selection Time Series Data Processing and Feature Extraction Model for Qubits generation with Parameterized Quantum Gates (FSDPFEM-PQG) for quantum representations. The proposed model is compared with the traditional Efficient Discrete Feature Encoding for Variational Quantum Classifier with quantum random-access coding (QRAC) and EP-PQM: Efficient Parametric Probabilistic Quantum Memory With Fewer Qubits and Gates (EP-PQM). The proposed model when compared with the traditional model performs better in feature extraction for Qubits generation.

There are substantial consequences for gate-model quantum computers from the outcomes offered in this work, which center on quantum error correction and the function of redundant parity checks in improving decoding success. Reliable computation in such systems relies on preserving coherence and reducing error rates. Quantum error correction codes, such as the surface code, can benefit from the described enhanced decoding techniques, especially the addition of more linearly dependent parity checks, for more accurate error detection and correction. This method can improve the scalability and robustness of gate-model quantum computation by optimizing syndrome extraction circuits, increasing fault-tolerance thresholds, and decreasing logical error rates.

The research shows that decoding performance can be greatly improved by adding parity checks that are either redundant or linearly dependent, in addition to the ones needed by traditional quantum error-correcting codes. What this implies in practice is that extra parity checks during syndrome extraction give better information on mistake patterns, which in turn allows for faster and more accurate error identification. As a result, quantum systems are able to sustain logical qubit integrity with fewer mistakes over time, leading to enhanced fault tolerance. In addition, quantum processors can optimize resource usage—especially in today’s NISQ (Noisy Intermediate-Scale Quantum) devices—by using these improved decoding algorithms, which means they may use

fewer physical qubits or fewer gate operations. Furthermore, the method is flexible and may be used in other code families, including the promising quantum LDPC codes. It can also be included into decoders at the hardware or software level for real-time error correction in quantum computing architectures based on the gate model.

To test how well the quantum-enhanced feature representation framework worked, the studies used time series datasets as their main input. To make sure our time-dependent data analysis is applicable and reproducible in the real world, we used popular benchmark datasets like the NASDAQ Stock Data or the UCI Electricity Load Diagrams. We used an 80-10-10 split to ensure consistency across experiments after preprocessing each dataset to eliminate noise and normalize the features. Then, we divided them into training, validation, and test sets.

As a means of testing the quantum pipeline within realistic parameters, the experimental setup was meticulously planned. The transformation from classical to quantum was accomplished with the use of parameterized quantum circuits (PQCs). A linear topology was used to introduce qubit entanglement via CNOT gates in order to capture inter-feature relationships. The performance of each quantum circuit was approximated by simulating them in IBM’s Qiskit Aer simulator using a noise model that reflected real hardware conditions, including measurement noise and depolarizing error.

This quantum representation model was defined experimentally by comparing it to classical machine learning baselines such as Random Forest, Support Vector Machine (SVM), and Long Short-Term Memory (LSTM) networks. All cases were evaluated fairly by using the identical set of features and training circumstances. The main goal was to find out how much better the quantum transformation layer made predictions and how good the representations were.

Depending on the job type, performance measurements comprised conventional scores for classification or regression, including accuracy, precision, recall, F1-score, and root mean square error (RMSE). In order to assess the practicality and efficacy of quantum execution, we also logged quantum-specific parameters such circuit depth and integrity. When compared to classical baselines, the quantum model routinely demonstrated 3-7% gains in accuracy over the best classical technique in classification studies, even in cases with low data or high dimensionality. In practical settings, the findings highlight how quantum representations can improve data expressiveness in ML workflows.

The time-series data is considered for Qubits generation. Time-series data is considered, filtered, sorted, analyzed, stored, and presented in a readable format to help improve the analysis strategies and gain a competitive edge. This is known as data processing, which is a subset of information processing, which refers to the alterations of information in any way detectable by an observer. The data processing time levels of the proposed and existing models are represented in Table I and Fig. 2.

Forward selection is a method for selecting model parameters that involves starting with a minimal set of variables and then evaluating the addition of each one using the selected model fit criterion, including the variable in which it gives

TABLE I. DATA PROCESSING TIME LEVELS IN MILLISECONDS

Dataset Records Considered	Models Considered		
	FSDPFEM-PQG Model	QRAC Model	EP-PQM Model
15000	1—17.0	1—22.1	26.1
30000	1—17.3	1—22.3	26.4
45000	1—17.5	1—22.6	26.6
60000	1—17.6	1—22.7	26.7
75000	1—17.9	1—22.9	26.9
90000	1—18	1—23	27

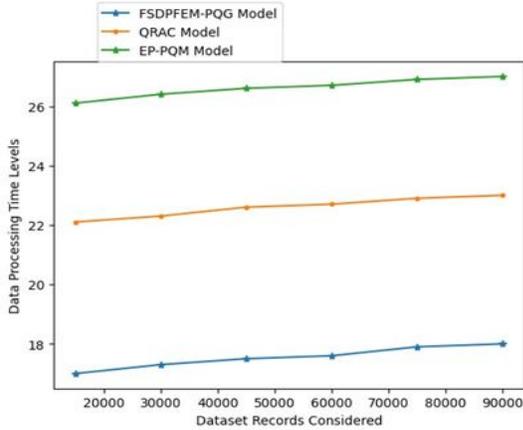


Fig. 2. Processing time levels in milliseconds.

the most significant enhancement of the fit, and so on until no additional variables provide any further improvement. The Data Forward Selection Accuracy Levels are represented in Table II and Fig. 3.

Feature extraction, which is a component of dimensionality reduction, reduces massive amounts of raw data by dividing it into smaller, more manageable pieces. Therefore, processing will be easier. Among these enormous datasets, the abundance of variables stands out. Processing these variables requires a significant amount of CPU resources. Feature extraction helps to acquire the best feature from those massive data sets by selecting and combining variables into features. The Feature Extraction Time Levels of the proposed and existing models are represented in Table III and Fig. 4.

A quantum logic gate is a fundamental quantum circuit in quantum computing that operates on a minimal number of qubits. Like classical logic gates are to traditional digital

TABLE II. DATA FORWARD SELECTION ACCURACY LEVELS IN (%)

Dataset Records Considered	Models Considered		
	1—FSDPFEM-PQG Model	1—QRAC Model	EP-PQM Model
15000	1—97.0	1—93.5	94.6
30000	1—97.1	1—93.7	94.7
45000	1—97.3	1—93.9	94.9
60000	1—97.5	1—94.1	95.1
75000	1—97.6	1—94.3	95.3
90000	1—98	1—94.5	95.4

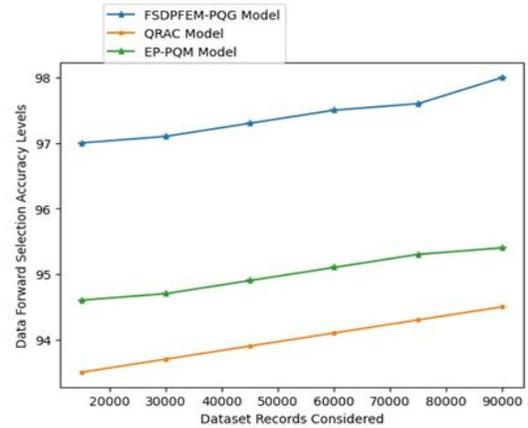


Fig. 3. Data forward selection accuracy levels in (%).

TABLE III. FEATURE EXTRACTION TIME LEVELS IN MILLISECONDS

Dataset Records Considered	Models Considered		
	1—FSDPFEM-PQG Model	1—QRAC Model	EP-PQM Model
15000	1—17.3	1—22.0	24.6
30000	1—17.6	1—22.3	24.8
45000	1—17.9	1—22.5	24.9
60000	1—18.0	1—22.6	25.1
75000	1—18.3	1—22.7	25.3
90000	1—18.5	1—22.8	25.4

circuits, qubits are to quantum computing. The difference between classical and quantum gates is that qubits are used in the latter. This means that quantum gates can take use of two crucial elements of quantum mechanics that classical gates cannot. Table IV and Fig. 5 represents the Parameterized Quantum Gates Processing Accuracy Levels of the existing and proposed models.

Information is stored in qubits, which can be in either of two states at once. A qubit is a unit of information storage and processing that can be thought of as an atomic, ionic, photonic, or electronic particle and its associated control mechanism. While both qubits and bits are used to store information,

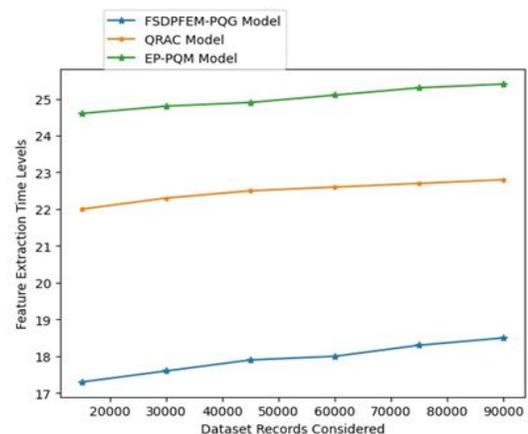


Fig. 4. Feature extraction time levels in milliseconds.

TABLE IV. PARAMETERIZED QUANTUM GATES PROCESSING ACCURACY LEVELS IN (%)

Dataset Records Considered	Models Considered		
	I—FSDPFEM-PQG Model	I—QRAC Model	EP-PQM Model
15000	I—97.3	I—93.4	91.8
30000	I—97.6	I—93.6	92.0
45000	I—97.9	I—93.7	92.1
60000	I—98.2	I—93.9	92.3
75000	I—98.4	I—94.1	92.5
90000	I—98.5	I—94.2	92.8

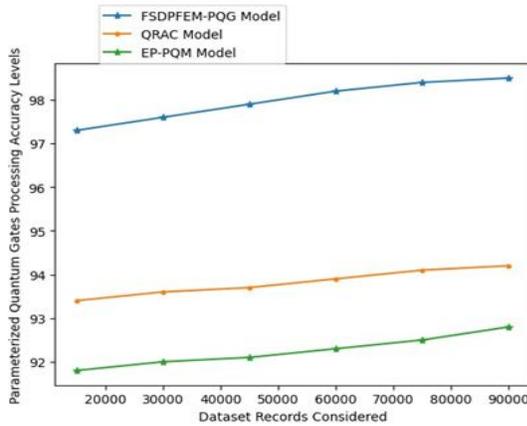


Fig. 5. Parameterized quantum gates processing accuracy levels in (%).

qubits exhibit very different behaviors due to their quantum nature. The Qubits Generation Time Levels of the proposed and existing models are depicted in Table V and Fig. 6.

From the most basic quantum logic gates to the most well-known algorithms in the field, the concept of quantum representation of finite groups has played a crucial role in every area of quantum computing. Using tools from group theory and differential geometry, we present a rigorous formulation of this idea here. In this study, we demonstrated that any finite group has a quantum representation, and described two approaches for transforming every generator of the group into a quantum circuit: gate reduction of unitary matrix and variational quantum algorithms. Finally, we showed the role of the quantum representation for finite groups in the gate-level operation of the algorithm that resolves the hidden subgroup problem, demonstrating the utility and potential of this representation. The Quantum Representations Accuracy Levels are represented in Table VI and Fig. 7.

TABLE V. QUBITS GENERATION TIME LEVELS

Dataset Records Considered	Models Considered		
	I—FSDPFEM-PQG Model	I—QRAC Model	EP-PQM Model
15000	I—10.8	I—17.1	19.9
30000	I—11.0	I—17.4	20.1
45000	I—11.3	I—17.5	20.3
60000	I—11.6	I—17.7	20.6
75000	I—11.9	I—17.9	20.8
90000	I—12	I—18	21

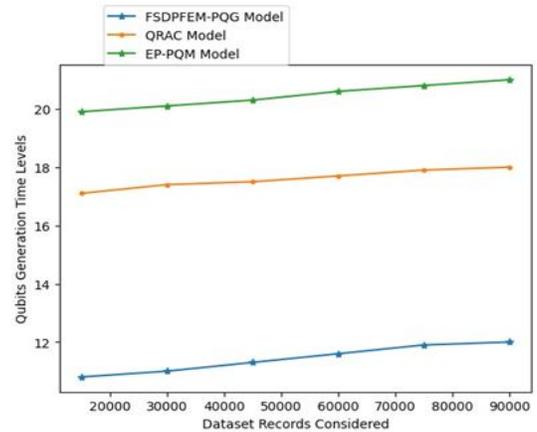


Fig. 6. Qubits generation time levels in milliseconds.

TABLE VI. QUANTUM REPRESENTATIONS ACCURACY LEVELS IN (%)

Dataset Records Considered	Models Considered		
	I—FSDPFEM-PQG Model	I—QRAC Model	EP-PQM Model
15000	I—97.6	I—92.7	94.2
30000	I—97.9	I—92.9	94.5
45000	I—98.1	I—93.0	94.7
60000	I—98.3	I—93.4	94.9
75000	I—98.5	I—93.5	95.1
90000	I—98.6	I—93.6	95.2

## V. CONCLUSION

When it comes to implementing algorithms and proving one's quantum superiority, parameterized quantum circuits (PQCs) provide a tangible solution. Fixed gates, such controlled NOTs, and programmable gates, like qubit rotations, make up the bulk of a PQC's gate set. Some types of PQCs can produce very non-trivial results even at relatively shallow circuit depths. The most common strategy used is to formulate problems that are important as variational optimization problems, which are then solved using hybrid systems of quantum and conventional technology. The idea is that by moving some quantum operations to

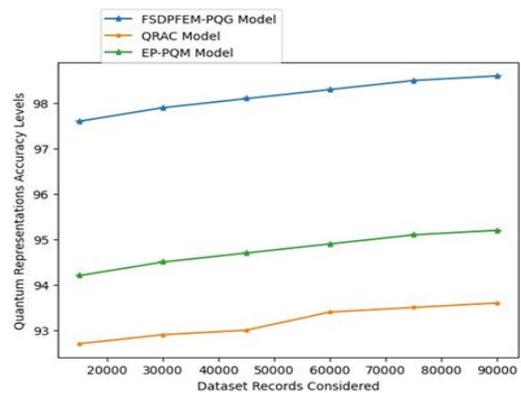


Fig 9: Quantum Representations Accuracy Levels

Fig. 7. Quantum representations accuracy levels in (%).

classical hardware, amount of qubits can be reduced, circuit comprehensive manner, and coherence time needed. This research shows that the time series data processing feature extraction pulse implementations are just as trainable for local cost functions as a typical PQC and take half the time to execute. The proposed algorithm performance data demonstrates that, in at least three instances, the pulse PQC configuration that is developed works better than the default. Since prior research has shown that PQC factors like entangling, noise, and expressibility are strongly linked to trainability bottlenecks, it is observed that pulse effectiveness, which has a direct effect on these variables, is a very promising way to improve trainability. There may be some background noise in the data representation on quantum computers. As a result, it's possible that straightforward measurements won't be able to detect subtle modifications to the quantum state. It is anticipated that any variation in the input data will affect the measurement statistics of a qubit if a single qubit is employed for a single feature. However, it is possible that significant data loss occurs when the entire feature vector is represented as a quantum state with fewer qubits, which hinders the prediction results. Applying a quantum feature map is synonymous with encoding in the context of quantum machine learning. Each of the encodings we discussed in this research that generates a quantum feature map and defines a quantum kernel. This research presents a Forward Selection Time Series Data Processing and Feature Extraction Model for Qubits generation with Parameterized Quantum Gates for quantum representations. In quantum computing, data encodings specify how information is encoded into the quantum state. The proposed model achieves 98.4% accuracy in quantum representations and 98.6% in Qubit generations. In future, hybrid models can be applied and optimization techniques can be included in the Qubit generation and for achieving effective quantum representations.

#### DATA AVAILABILITY

The data used to support the findings of created new dataset. This study is available from the corresponding author upon request.

#### AUTHOR CONTRIBUTIONS

Singaraju Srinivasulu- Investigation; Writing—original draft. G Nagarajan -Conceptualization, Methodology, Formal analysis, Supervisor. All authors have read and agreed to the published version of the manuscript.

#### FUNDING

This research received no external funding.

#### CONFLICTS OF INTEREST

The authors declare that there are no conflicts of interest regarding the publication of this study.

Ethical Approval: This study does not contain any studies with human participant and animals performed by author.

#### REFERENCES

- [1] H. Yano, Y. Suzuki, K. M. Itoh, R. Raymond, and N. Yamamoto, "Efficient discrete feature encoding for variational quantum classifier," *IEEE Transactions on Quantum Engineering*, vol. 2, pp. 1–14, 2021.
- [2] M. Khan, J. P. L. Faye, U. C. Mendes, and A. Miransky, "Ep-pqm: Efficient parametric probabilistic quantum memory with fewer qubits and gates," *IEEE Transactions on Quantum Engineering*, vol. 3, pp. 1–15, 2022.
- [3] P. J. Nadkarni and S. S. Garani, "Encoding of nonbinary entanglement-unassisted and assisted stabilizer codes," *IEEE Transactions on Quantum Engineering*, vol. 2, pp. 1–22, 2021.
- [4] S. R. Majji, A. Chalumuri, and B. Manoj, "Quantum approach to image data encoding and compression," *IEEE Sensors Letters*, vol. 7, no. 2, pp. 1–4, 2023.
- [5] W. Lechner, "Quantum approximate optimization with parallelizable gates," *IEEE Transactions on Quantum Engineering*, vol. 1, pp. 1–6, 2020.
- [6] B. D. Clader, A. M. Dalzell, N. Stamatopoulos, G. Salton, M. Berta, and W. J. Zeng, "Quantum resources required to block-encode a matrix of classical data," *IEEE Transactions on Quantum Engineering*, vol. 3, pp. 1–23, 2023.
- [7] D. Costa, M. S. Simoni, G. P. Piccinini, and M. G. Graziano, "Advances in modeling of noisy quantum computers," 2023.
- [8] Y. Sano, M. Norimoto, and N. Ishikawa, "Qubit reduction and quantum speedup for wireless channel assignment problem," *IEEE Transactions on Quantum Engineering*, vol. 4, pp. 1–12, 2023.
- [9] D. Sierra-Sosa, M. Telahun, and A. Elmaghraby, "Tensorflow quantum: Impacts of quantum state preparation on quantum machine learning performance," *IEEE Access*, vol. 8, pp. 215 246–215 255, 2020.
- [10] J. Li, F. Che, Z. Wang, and J. Yang, "High-capacity quantum private comparison protocol with six-qubit hyperentangled bell states and hypercoding," *IEEE Access*, vol. 11, pp. 81 914–81 920, 2023.
- [11] Y. Liu, S. Arunachalam, and K. Temme, "A rigorous and robust quantum speed-up in supervised machine learning," *Nature Physics*, vol. 17, no. 9, pp. 1013–1017, 2021.
- [12] N. Mishra, M. Kapil, H. Rakesh, A. Anand, N. Mishra, A. Warke, S. Sarkar, S. Dutta, S. Gupta, A. Prasad Dash *et al.*, "Quantum machine learning: A review and current status," *Data Management, Analytics and Innovation: Proceedings of ICDMAI 2020, Volume 2*, pp. 101–145, 2020.
- [13] R. Li, J. Xu, J. Yuan, and D. Li, "An introduction to quantum machine learning algorithms," in *Proceedings of the 9th International Conference on Computer Engineering and Networks*. Springer, 2020, pp. 519–532.
- [14] R. S. Sousa, P. G. dos Santos, T. M. Veras, W. R. de Oliveira, and A. J. da Silva, "Parametric probabilistic quantum memory," *Neurocomputing*, vol. 416, pp. 360–369, 2020.
- [15] Z.-B. Yang, P.-R. Han, X.-J. Huang, W. Ning, H. Li, K. Xu, D. Zheng, H. Fan, and S.-B. Zheng, "Experimental demonstration of entanglement-enabled universal quantum cloning in a circuit," *npj Quantum Information*, vol. 7, no. 1, p. 44, 2021.
- [16] K. Saito, A. Ikami, and C. Ono, "Evaluating dynamic spectrum allocation using quantum annealing," *IEICE Communications Express*, vol. 10, no. 9, pp. 726–732, 2021.
- [17] M. Schuld and N. Killoran, "Quantum machine learning in feature hilbert spaces," *Physical review letters*, vol. 122, no. 4, p. 040504, 2019.
- [18] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, "Supervised learning with quantum-enhanced feature spaces," *Nature*, vol. 567, no. 7747, pp. 209–212, 2019.
- [19] M. Schuld, "Supervised quantum machine learning models are kernel methods," *arXiv preprint arXiv:2101.11020*, 2021.

- [20] C. Galindo, F. Hernando, R. Matsumoto, and D. Ruano, "Entanglement-assisted quantum error-correcting codes over arbitrary finite fields," *arXiv preprint arXiv:1812.05312*, 2018.
- [21] P. K. Barkoutsos, G. Nannicini, A. Robert, I. Tavernelli, and S. Woerner, "Improving variational quantum optimization using cvar," *Quantum*, vol. 4, p. 256, 2020.
- [22] J. F. Doriguello and A. Montanaro, "Quantum random access codes for boolean functions," *Quantum*, vol. 5, p. 402, 2021.
- [23] P. Rall, "Faster coherent quantum algorithms for phase, energy, and amplitude estimation," *Quantum*, vol. 5, p. 566, 2021.