
Taylor & Francis
Taylor & Francis Group

IETE Journal of Research

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/tijr20

# Brain Tumour Classification Using Quantum Support Vector Machine Learning Algorithm

**Tarun Kumar, Dilip Kumar & Gurmohan Singh**

**To cite this article:** Tarun Kumar, Dilip Kumar & Gurmohan Singh (2024) Brain Tumour Classification Using Quantum Support Vector Machine Learning Algorithm, IETE Journal of Research, 70:5, 4815-4828, DOI: 10.1080/03772063.2023.2245350

**To link to this article:** https://doi.org/10.1080/03772063.2023.2245350

Published online: 13 Aug 2023.

IETE JOURNAL OF RESEARCH
2024, VOL. 70, NO. 5, 4815–4828
https://doi.org/10.1080/03772063.2023.2245350

# Brain Tumour Classification Using Quantum Support Vector Machine Learning Algorithm

Tarun Kumar [1], Dilip Kumar [1] and Gurmohan Singh [2]

[1] Electronics and Communication Engineering Department, Sant Longowal Institute of Engineering and Technology, Longowal, PB, 148 106, India; [2] Cyber Security Technology Division (CSTD), Centre for Development of Advanced Computing (C-DAC), Mohali, PB, 160 071, India

**ABSTRACT**

Quantum computing is an emerging field that can effectively solve several machine learning problems. The immensely growing data size has begun to create barriers for classical machine learning algorithms. Quantum computers proficiently handle and process big data composed of metrics and vectors. This paper presents the quantum machine learning model based on quantum support vector machines (QSVM) to classify brain tumours into malignant and benign. The dataset used in this experiment is Brats 2015, available on Kaggle. The QSVM-based model was run on real-time quantum machines and simulators to extract the performance metrics in terms of accuracy and execution time. The QSVM classification model is compared with its classical SVM equivalent. The kernel-based QSVM model implementation on a 32-qubit quantum simulator was 188 times faster with 95% accuracy which is 1.60% better than the classical equivalent. Similarly, the model takes 24.19% less time than the classical SVM model when realized on a 5-qubit real-time superconducting processor with the same accuracy. Hence, the results reveal that quantum machine learning models implemented on quantum computers outperformed their classical equivalents.

**KEYWORDS**

Brain tumour; Classifiers; Quantum machine learning; Quantum support vector machines; Quantum feature mapping; SVM

## 1. INTRODUCTION

Machine learning completely transformed the way of humans interact with data. It makes the systems learn and improve their experiences without programming them explicitly. Machine learning, being a fragment of statistics and Artificial Intelligence (AI), offers processing techniques for datasets for tasks such as classification, image segmentation, pattern identification, *etc.* [1]. Internet of Things (IoTs) devices, smart systems, social media, *etc.* contribute to the rapid growth of data. The size of data stored globally is increasing per year by almost 20%. This exponential growth of the data size introduces a new term big data [2]. Big data is a form of data which has high volume, large variety and high velocity. To process and analyze such datasets, their complexity creates barriers to conventional processing methods. Thus, the demand to improvise machine learning is escalating for the processing and management of big data [3,4]. Scientists, researchers, and some leading IT companies such as IBM, Google, Microsoft, *etc.* harness the power of quantum computing for the optimization of classical machine learning algorithms [2].

Quantum computing offers a different perspective on information processing. Unlike classical computers, which are constructed by physically implementing two states *i.e.* 1 and 0, quantum computers operate using quantum states $|0\rangle$, $|1\rangle$ and their linear combinations [5]. The fundamental unit of representing information in quantum systems is referred to as Quantum-bit (Qubit). Quantum computers gained the power to manipulate and handle multiple states simultaneously from the quantum mechanic principle of superposition and entanglement [6]. Computations executed by following the laws of quantum mechanics are generally called quantum computing. Quantum computing offers the manipulation of information with the help of quantum algorithms/circuits [6,7].

The amalgamation of machine learning and quantum computing has given rise to a new field named Quantum Machine Learning (QML) [8]. QML has the potential to revolutionize the field of artificial intelligence by exploiting the unique properties of quantum computing, such as superposition and entanglement, to enable faster and more accurate computations [9]. Although QML is often associated with big data, its applications are not limited to this domain alone. QML algorithms can solve certain problems that are difficult for classical algorithms, leading to more accurate predictions and faster processing times [10]. Quantum classification algorithms, such as quantum support vector machines (QSVM), quantum neural networks (QNN), and quantum k-nearest neighbours (QKNN) have been developed as potential solutions to problems that are difficult or impossible for classical algorithms to solve [9,10]. These algorithms have promising applications in computer-aided diagnosis (CAD), where they can help healthcare professionals make more accurate and timely diagnoses. Also, CAD eliminates the subjective bias of diagnosis [11]. CAD is rapidly gaining popularity in the healthcare sector including the classification of brain tumours, which are abnormal growths of tissue in the brain that can be either benign or malignant. Classification of brain tumours involves using computer algorithms to assist physicians in accurately identifying and classifying brain tumours based on medical imaging data [12]. The CAD process involves image pre-processing, image segmentation, feature extraction, and classification steps to detect and classify brain tumours into benign and malignant categories [13,14]. QML models for classification use quantum algorithms to classify data points by converting classical data points to quantum states using quantum feature maps, processing them with a quantum classifier circuit, and utilizing the results to make predictions [8–10]. With the aid of QML, QSVM can be utilized to classify brain tumours into malignant and benign.

This paper reports the development and implementation of a quantum machine learning model for classification in brain tumour dataset. A QSVM algorithm with two different approaches is used to develop quantum machine learning models by constructing the quantum feature map circuit with the appropriate repetitions which produce the results with better accuracy in less execution time. The performance of the QSVM-based classification models for the brain tumour dataset is compared with the SVM-based classification model. The paper is structured as follows: Section 2 introduces the data classification using classical machine learning. Section 3 reports the outline of data classification using quantum machine learning. Section 4 presents the related work in the field of quantum machine learning and brain tumour classification using classically developed models. Section 5 reports the proposed methodology for the development of the QML-based classification model. Section 6 presents experimental results and a comparison of QSVM-based classification models with SVM-based models. Section 7 concludes the paper.

## 2. DATA CLASSIFICATION USING CLASSICAL MACHINE LEARNING

Machine learning categorizes data into classes which can be obtained from the structure of data or can be defined by the user [15]. It is classified as supervised [15], unsupervised [16], and reinforced [17] machine learning. Supervised learning and unsupervised learning are based on learning from the data as it involves data analysis and mining, whereas reinforced learning is based on learning from interactions and improves the learning model at each step [18].

A dataset $X = x_1, x_2, x_3, \ldots, x_n$ is taken to understand the concept of machine learning, where $x_i$ denotes the number of data points in the dataset. Dataset $X$ is divided into training data ($X_T$) which are labelled and test data ($X_0$) which are unlabelled. Supervised machine learning constitutes a set of already defined training data $X_T$ comprising of already classified data points to generate a set of classifications $Y = y_1, y_2, y_3, \ldots, y_n$, where $y_i$ denotes the class for data points $x_i$. $X_T$ and $Y$ both are fed to a machine learning algorithm which optimizes their internal parameters up to a point when the training data are classified into the nearest value of $Y$. This is done by adjusting model parameters to minimize a loss function that measures the difference between predicted and true outputs. The time when the machine learns completely, unlabelled data $X_0$ are given to the machine for classification and the machine predicts the output for $X_0$ [15]. Supervised machine learning algorithms generally take care of classification and regression problems. In the case of unsupervised machine learning, the classification set or class is not defined *i.e.* $Y$ is inexistent. It happens because of the high complexity of the dataset. Learning of machines in such cases is based on the internal structure of input data [9]. Algorithms fall under the category of unsupervised learning taking unlabelled training data $X_T$ (unlabelled in the case of unsupervised learning) as input and looking for hidden structures in it [15]. Such algorithms contribute to solving dimensionality reduction and clustering problems [18]. Sometimes high data complexity makes it difficult for a machine to predict a desired output in the case of supervised and reinforced machine learning. In such cases, data can be separated into different clusters by using clustering algorithms and further can be classified [15]. Reinforced machine learning, on the other hand, is a category that falls in between supervised and unsupervised machine learning as the instant output corresponding to input is not correct but some sort of supervision exists [9]. Instead of getting desired output corresponding to the input, the agent learns based on the feedback received from the environment. The feedback includes rewards and punishment based on the actions and experiences of the agent. It is helpful to algorithms/machines as it reveals if the output is improved or degraded based on the steps chosen in the learning process. The application of reinforced learning is to train a model to control self-driving automobiles and in video games where computers play with humans as opponents [2,9].

Machine learning offers numerous algorithms such as K-nearest neighbours [19], Decision tree [20], Support Vector machines [21], Naive Bayes [22], *etc.* for classification. Support Vector Machine algorithm that belongs to supervised learning is used in classification problems. The task of SVM is searching a hyperplane, which separates two classes of data and the margin between the support vectors of these two classes, should be maximum. Maximum margin helps SVM to classify efficiently [21]. The hyperplane is nothing but a decision boundary responsible for the classification of two classes. The data points closest to the hyperplane are the support vectors which play a major role in the alteration of orientation and placement of the hyperplane. The classification of data in SVM is done in two classes with values 1 and $-1$ [10].

Assuming a training data $(x_1, y_1), (x_2, y_2), \ldots\ldots, (x_n, y_n)$, such that $y \in \{1, -1\}$ are the two classes which represent the data as $+1$ for one class and $-1$ for the second class. Such data can be simply divided by using a hyperplane, generally represented as $w^T x - b = 0$, where $\vec{w}$ signifies a vector normal to the hyperplane and $b$ represents the bias parameter [23]. Assuming the data are linearly separable, SVM constructs two hyperplanes parallel to each other to draw a hard margin and the hyperplanes are distant from each other by $\left|\left|\frac{2}{w}\right|\right|$. The hyperplanes are constructed in such a way that $w^T x - b \leq 1$ for $y_{i=1,2\ldots,N} = -1$ and $w^T x - b \geq 1$ for $y_{i=1,2,\ldots,N} = 1$. The above conditions for hyperplanes can be written as $y_i(w^T x - b) \geq 1$ [10,21]. This hyperplane separates the data points into two classes, as depicted in Figure 1.

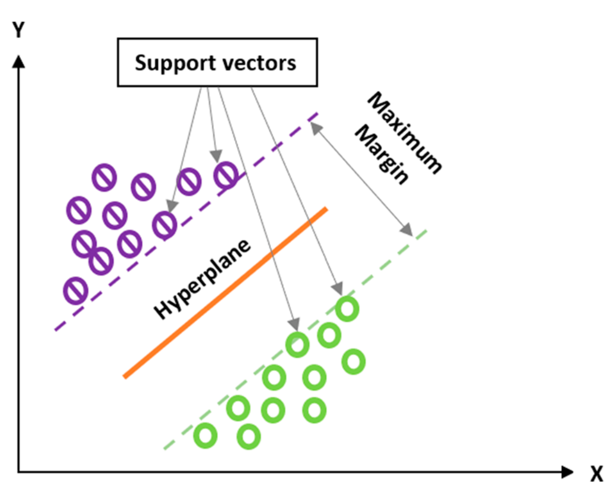



**Figure 1:** Relationship between hyperplane and support vectors

Most of the time, data are not linearly separable, hence the above-mentioned method fails to separate the data points. The simplest way to tackle such a problem is using the kernel trick which maps the problem to a higher dimension. The problem will be solved by simply drawing a hyperplane in the higher dimension [24]. To achieve this, Lagrangian multipliers ($\alpha$) are used to optimize dual formulation which leads to the modification of the expression to $L(\vec{\alpha}) = \sum_{i=1}^{N} y_i \alpha_i = \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j K_{ij}$ with constraints $\sum_{i=1}^{N} \alpha_i = 0$ and $y_i \alpha_i \geq 0$. $\vec{w}$ and $b$ are the hyperplane parameters that can be calculated using $\vec{w} = \sum_{i=1}^{N} \alpha_i x_i = \frac{1}{2}$ and $b = y_i - \vec{w} x_i$ [25]. Here, $K_{ij}$ represents the kernel matrix and $K_{ij}(\overrightarrow{x_i}, \overrightarrow{x_j}) = \overrightarrow{x_i} . \overrightarrow{x_j}$ is the inner product of the data points in high dimensional space. The function to decide the hyperplane becomes $f(x) = sgn\left(\sum_{i=1}^{N} \alpha_i y_j K(x_i, x) + b\right)$. Hence, SVM can solve non-linear classification problems also. In addition to that regularization, it is utilized in SVMs to prevent overfitting, which arises when the model becomes too complex and fits the training data too closely, resulting in poor generalization performance on new data [9].

## 3. DATA CLASSIFICATION USING QUANTUM MACHINE LEARNING

QML lies at the intersection of ML and quantum computing. [8]. The data processing in quantum machine learning is done on quantum computers. Quantum algorithms are generally derived from classical algorithms so that they can be used on quantum computers. The methods like deep neural networks in classical machine learning can identify the statistical patterns present in data and are capable of producing data of the same patterns [9,10]. It is observed that if information processed in quantum systems/algorithms produces statistical patterns which classical computers find hard to produce, then surely quantum algorithms can identify the patterns which classical computers failed to identify easily [9]. QML algorithms utilize both unsupervised and supervised learning techniques. In quantum algorithms, data are mapped in qubits followed by unitary operations on them and finally deliver the output upon measuring the state of qubits [23].

Data classification is a vital tool offered by machine learning. The classification tools are extensively used in face detection [26], image classification [27], bioinformatics [28], handwriting recognition [29], *etc.* These classification tools generally have machines which could recognize and deal with the data. The widely used classification tool is the SVM algorithm in the context of classical machine learning and the quantum version of SVM *i.e.* the QSVM algorithm in QML [30]. The SVM algorithm explicitly handles the non-linear classification problems using kernels. But many times, a huge number of dimensions are required to solve a problem. It leads to increased complexity and overfitting because of the expansion of a sparse matrix specifying the position of data points and referred to as the Curse of Dimensionality. This is when quantum SVM comes into play as classical computers find it difficult to solve problems in higher dimensions [9]. The QSVM algorithm offers exponential speedup in solving a problem when implemented on the quantum computer. The variational QSVM computes the hyperplane(s) for the classification of new test data. It can solve problems which require classification in more than two classes. In this approach, two quantum algorithms are used such that the first algorithm finds the hyperplane using training data whereas the second one classifies the new test data [31]. The kernel-based QSVM approach requires only one algorithm and comes in use generally in the case of binary classification tasks. To classify test data, a quantum machine computes the kernel matrix from training data followed by the computation of support vectors using a classical machine [9,31]. In quantum computing, the algorithms have some liberty to misclassify the data to deal with overfitting and slack variable $\xi_i$ measures the misclassification provided $\xi_i \geq 0$. The optimization problem can be written as (1)

$$min\left[\frac{1}{2}\|w\| + C\sum_{i=1}^{N}\xi_i\right] \tag{1}$$

where the cost parameters C and $\gamma$ are related as $C = \frac{\gamma}{2}$. The above expression further transformed into $\overrightarrow{w}.\overrightarrow{x_i} - b = 1 - \xi_i$ and upon using least square approximation the problem can be written as (2)

$$F\begin{pmatrix} b \\ \vec{\alpha} \end{pmatrix} = \begin{pmatrix} 0 & \vec{I}^T \\ \vec{I} & K + \gamma^{-1}I \end{pmatrix}\begin{pmatrix} b \\ \vec{\alpha} \end{pmatrix} = \begin{pmatrix} 0 \\ \overrightarrow{y_i} \end{pmatrix} \tag{2}$$

where $I$ and $\overrightarrow{y_i}$ signify the unit matrix and training data labels, respectively [9,30]. The value of the SVM classifier depends upon the constraints $\vec{\alpha}$ and $b$. The classification of new datapoint $\overrightarrow{x_0}$ can be done as (3)

$$\begin{aligned} y_i(\overrightarrow{x_0}) &= sgn(\vec{w}\overrightarrow{x_0} + b) \\ &= sgn\left[\sum_{i=1}^{N}\alpha_i k(\overrightarrow{x_i}\,\overrightarrow{x_0}) + b\right] \end{aligned} \tag{3}$$

Classically, SVM can solve a problem in $o(\log(\varepsilon^{-1})poly\,(N, M))$ time, where $\varepsilon$, $N$, and $M$ signify the accuracy, dimensionality index, and several training vectors, respectively. The time taken by the classical SVM for handling several vectors and the dimensionality of the vector space is polynomial. On the other hand, quantum SVM takes logarithmic time for the same task [30]. First, quantum data are obtained by mapping the classical data into quantum states and then QSVM processes the quantum data using methods such as Quantum Fourier Transform (QFT) and matrix inversion. Quantum SVM solves a problem in $o(log_2(N, M))$ that means it offers an exponential speedup compared to its classical equivalents. One more advantage is that while performing classification tasks on quantum computers, Quantum Random Access Memory (QRAM) stores the quantum states which can be accessed parallelly during the processing and leads to exponential speedup [32].

## 4. RELATED WORK

Many researchers have investigated the efficacy of Quantum Support Vector Machines based on their theoretical and practical implementations for quantum machine learning problems. J. Biamonte *et al.* [8] explained how the crossover of quantum computing and machine learning benefitted the development of quantum machine learning algorithms. Machine learning enhanced the benchmarking and control of the quantum systems by following the principles of quantum mechanics. It leads to the performance improvement of the quantum systems by reducing the computational complexity. They have discussed various quantum techniques to process the big data which include Grover's algorithm for amplitude amplification, QSVM for classification tasks, k-means clustering for clustering tasks, *etc.* The authors highlighted that these algorithms offer quantum speedup. J. C. Adcock *et al.* [15] presented an overview of classical machine learning and quantum machine learning, compared both classical and quantum machine learning and principal component analysis. The authors also discussed the quantum algorithm which includes the HHL algorithm for solving a linear system of equations, the k-Nearest Neighbour (KNN) algorithm, the SVM algorithm, *etc.* and implemented a QSVM on a four-qubit quantum simulator to check whether a hand-written number is 6 or 9. Nimish Mishra *et al.* [9] explained the influence of quantum computers on normal processing and machine learning. The authors discussed algorithms such as quantum HHL, SVM, QSVM, *etc.* The implementation techniques, applications of quantum algorithms, and the challenges such as data handling and data visualization in QML are discussed. S. Saini *et al.* [23] presented a classification model based on QSVM and implemented it on the breast cancer dataset. They revealed that due to the complex computations performed on quantum computer/simulator, QSVM lagged in accuracy compared to SVM but the computational speed offered by the quantum simulator is 234 folds quicker than its classical equivalent. R.D.M. Simoes *et al.* [33] explored the application of quantum machine learning in solving practical problems, focusing on kernel-based quantum support vector machines and quantum neural networks. By evaluating these algorithms on five different datasets with various quantum feature maps, the experiments show that quantum support vector machines have an accuracy improvement of 3%–4% over classical solutions on average. Although the experiments were conducted on relatively small datasets, the results demonstrate the potential of quantum computing in solving small-scale machine learning problems with better accuracy and less complexity.

S.S. Kavitha *et al.* [34] examined the execution speed and accuracy of QSVM classification compared to classical SVM classification by selecting the appropriate quantum feature mapping. To achieve optimal classification performance on some complex datasets (wine, HCV, electrical grid), the authors utilized QSVM. The study focuses on selecting the best feature map for benchmark datasets and experimental results show that the processing time of QSVM-based models are considerably reduced compared to SVM-based models. G. Singh *et al.* [35] examined the potential of quantum machine learning algorithms for pattern classification using a benchmarking MNIST dataset. The authors claimed that the quantum classifiers exhibited a significant speed advantage over their classical counterparts, with computational efficiency up to 81.62% higher when implemented on real 5- and 14-qubit quantum processors. Furthermore, the quantum SVM kernel matrix algorithm showed 6.4% better accuracy than classical SVM.

Since very less work in the field of brain tumour classification using QML is reported, some latest developed QML-based models and classical models are explored. J. Amin *et al.* [36] introduced a method for detecting brain tumours that employ deep features derived from the Inceptionv3 model and a quantum variational classifier (QVR). The QVR is leveraged to differentiate between different tumour types in MRI images. The model's effectiveness is evaluated on three benchmark datasets and locally collected images. The results demonstrate that the proposed model achieved a detection score of 90.9%, highlighting its efficacy for brain tumour detection. E. Akpinar *et al.* [37] compared the efficacy of SVM and QSVM in differentiating medulloblastoma (MB) and ependymoma (EP) using multiparametric MRI. The authors analyzed the MRI of 49 children, including 40 with MB and 9 with EP, and utilized semi-quantitative MR imaging features to evaluate the vascularity characteristics of the lesion. The authors used a precomputed quantum kernel and fed it into a classical SVM for tumour classification. The QSVM deliver similar results as SVM with a test score value of 90% for all kernels except for the poly kernel, whose test score is 80%.

W. Ayadi *et al.* [38] presented a hybrid optimization algorithm for brain tumour segmentation and classification by considering the extracted features as the input of the deep residual network using the proposed chronological Jaya honey badger algorithm. This method achieved an accuracy of 95% for brain tumour classification. N.S. Shail *et al.* [39] presented a classification model using a multi-level attention network (MANet) for brain tumour classification. The MANet includes spatial and cross-channel attention which not only focuses on prioritizing tumour regions but also maintains cross-channel temporal dependencies in semantic feature sequences obtained from the Xception backbone. This method achieved an accuracy of 94.9% for brain tumour classification on the Brats 2018 dataset. R. Hao *et al.* [40] developed a transfer learning–based active learning framework to reduce the annotation cost while maintaining the stability and robustness of the model performance for brain tumour classification. The authors integrate the traditional uncertainty sampling technique and query-by-committee method with transfer learning to reduce the number of required training samples while maintaining the stability and robustness of convolutional neural network performance for brain tumour classification. The proposed method achieved an accuracy of 82.8% for brain tumour classification. M. Mudda *et al.* [41] presented a system that makes the image fusion of three MRI images of the same patient taken at different angles and with different points by extracting Gray Level Run Length Matrix and Center-Symmetric Local-Binary Pattern features to classify the tumour. This system can classify the tumour as a benign tumour, or a malignant tumour with 94% accuracy.

The QML is still in the early stages of development and is considered the next big thing in computing due to its exceptional computational abilities. Therefore, it is essential to explore the performance of quantum computing and QML in every field, including the healthcare sector. The healthcare sector has yet to fully explore the potential of quantum computing and QML and to contribute to the related work, a QSVM-based model is proposed for brain tumour classification.

## 5. PROPOSED METHODOLOGY

The objective of this paper is to classify brain tumours as malignant or benign using the QSVM algorithm. To initiate the classification process, a dataset is created from brain MRI images (Brats 2015) through pre-processing, image segmentation, and feature extraction techniques. The features extracted from the brain MRI images are utilized to form the dataset for classification. Since the brain tumour dataset in.csv format is readily accessible online [42], it is downloaded directly to address the classification problem. The process of developing a QML-based classification model for the brain tumour dataset comprises several steps, such as dataset pre-processing, exploratory data analysis (EDA) and dataset visualization, splitting the dataset into train data, and test data, principal component analysis (PCA), quantum feature mapping (QFM), defining quantum kernel for QSVM, defining quantum instance, initializing QSVM, training the classification model using train data, and predicting with test data and analyzing the results. The classification process for brain tumours is depicted in Figure 2.

### 5.1 Dataset

The dataset was obtained from the Kaggle repository of Jakesh Bohaju [42]. A total of 3763 rows and 15 columns are present in the dataset. After removing the headers and labels the dataset consists of 3762 samples (class 0 = 2079 and class 1 = 1683) and 13 features. The brain tumour dataset contains five first-order and eight second-order features extracted from the brain tumour MRI images [42]. The first-order features are mean, variance, standard deviation, skewness, and kurtosis and the second-order features are contrast, energy, angular second movement (ASM), entropy, homogeneity, dissimilarity, correlation, and coarseness. All the features present in the dataset reveal information about the patient's brain MRI images.

### 5.2 Data pre-processing, visualization, and EDA

The initial stage of QML-based classification involves dataset pre-processing, which comprises a range of activities such as formatting, normalization, rescaling, and cleansing, and requires the data to be visualized to create an efficient classification model. Pair plots are utilized to visualize the association between the features of the brain tumour dataset. Once the data are visualized, EDA [43] is carried out to obtain a thorough understanding of the dataset. The dataset is scanned for missing or null entries, and the statistical summary of the dataset is extracted to identify any outliers or extreme values. Furthermore, the

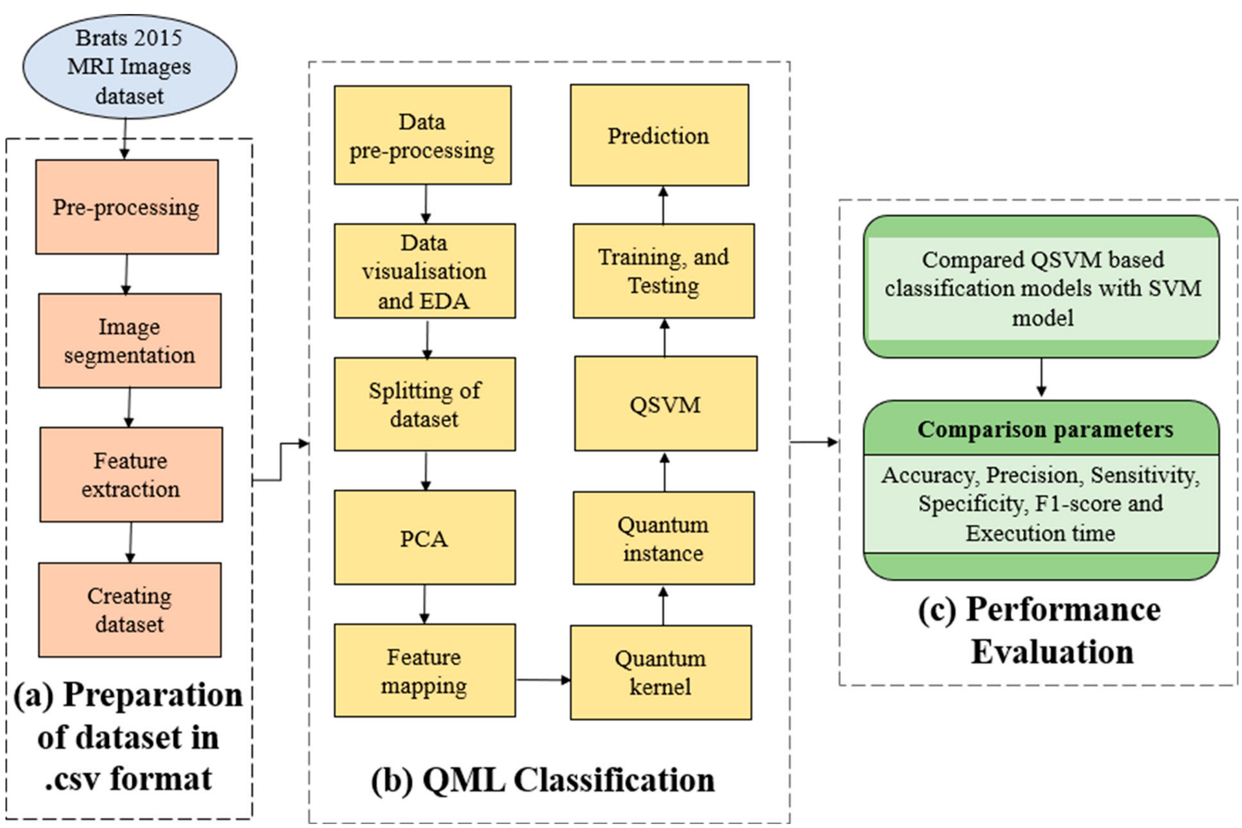



**Figure 2:** Proposed methodology

unnecesary image column is eliminated from the dataset. Finally, the dataset is randomly split into the training set (70%, $n = 2633$) and the testing set (30%, $n = 1129$).

## 5.3 PCA

The number of qubits is set to 13, which is equivalent to the number of features in the brain tumour dataset which makes the quantum circuit of the classifier complex. However, to reduce the complexity of the circuit and the dataset, PCA [44] is employed. This approach aids in reducing the dimensionality of the dataset while retaining feature-based information. PCA reduces the number of features from 13 to 2.

## 5.4 QFM

To encode the data points of the classical brain tumour dataset into quantum states, a quantum feature map is used, which maps classical data $\vec{x}$ to a quantum state $|\varphi(\vec{x})\rangle$in the quantum Hilbert space. The feature map is represented by unitary as $U_{\varphi(\vec{x})} = U_{\varphi(\vec{x})} \otimes H^n U_{\varphi(\vec{x})} \otimes H^n$ for n-qubits, where H is the Hadamard gate. Quantum feature mapping is a challenging task. The challenge is to create the quantum circuit of the feature map with an appropriate number of repetitions to deliver better performance in terms of accuracy and execution time. Various quantum feature maps [45,46] such as PauliFeatureMap, ZFeatureMap, and ZZFeaturemap can be used to map brain tumour dataset features to quantum states. In this experiment, the ZZFeaturemap circuit with 2 repetitions is constructed to map classical data points of the brain tumour dataset to quantum states after PCA which delivers better results than other Pauli feature maps (X, Y, Z, XX, XY, XZ, YY, YX, YZ, ZX, and ZY). All these Pauli feature maps were tested with different numbers of repetitions but the ZZFeaturemap circuit with 2 repetitions gives the best results in terms of accuracy and execution time. It is worth noting that the depth and width of the quantum feature map circuit should be less because larger width and depth increase the complexity of the circuit, which makes it error-prone when implemented on real-time quantum computers. The ZZFeaturemap circuit with two repetitions is employed and illustrated in Figure 3.

## 5.5 Quantum kernel

The notion of defining a quantum kernel is analogous to that of a classical kernel, which projects the data points of a brain tumour dataset into a higher-dimensional feature space of dimensionality $m$. In the case of classical ML, $k$, a kernel function, is employed to calculate the inner product of the n-dimensional input data points $\overrightarrow{x_i}, \overrightarrow{x_j}$ as $k(\overrightarrow{x_i}, \overrightarrow{x_j}) = \langle f(\overrightarrow{x_i}), f(\overrightarrow{x_j})\rangle$, where f maps the data points into the higher-dimensional space. The quantum kernel $K$ [47] computes the inner product of quantum feature maps, $K(\vec{x}, \vec{z}) = |\langle\varphi(\vec{x})|\varphi(\vec{z})\rangle|^2$.

## 5.6 Quantum Instance

QML utilizes QISKIT [48], an IBM Quantum library, to construct and run quantum circuits/algorithms. While classical SVM operates on a local CPU environment, QSVM requires QISKIT. The quantum instance comprises Qiskit terra backend and configuration for transpiling and executing the quantum circuit. This experiment employs three IBMQ backends, namely *statevector_simulator* [49], *ibmq_qasm_simulator* [50], and real-time quantum computer *ibmq_lima* [51]. The QSVM uses the quantum instance to execute the quantum circuits on these backends. The local machine used to run the SVM-based classification model and to access the quantum backends runs on Windows 11, with an i5 processor, 8 GB RAM, and 512 GB SSD. Table 1 provides the characteristics of the IBM backends used in this experiment.

## 5.7 QSVM-based Classification

To develop a classification model for the brain tumour dataset, QSVM is applied after defining the quantum

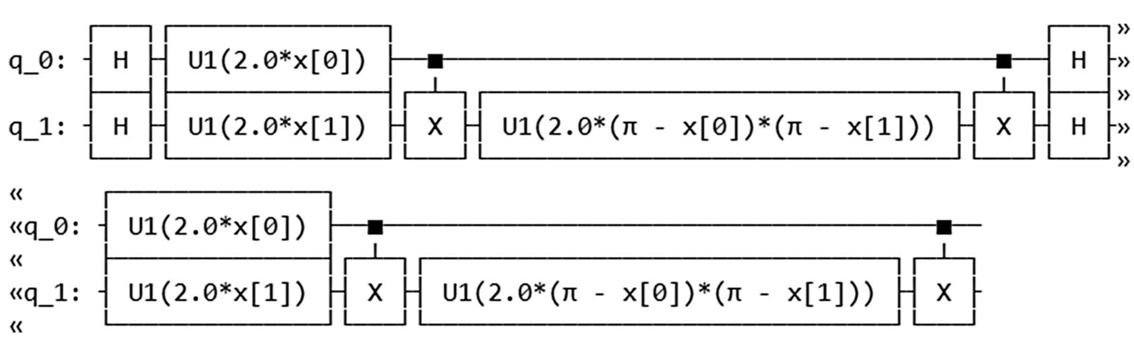



**Figure 3:** ZZFeaturemap with 2 repetitions

**Table 1: Characteristics of IBM backends [49–51]**

| Characteristics | ibmq_qasm_ simulator | statevector_ simulator | ibmq_lima |
|---|---|---|---|
| Number of qubits | 32-qubit | 32-qubit | 5-qubit |
| Quantum volume | N/A | N/A | 8 |
| CLOPS | n/A | N/A | 2.7 K |
| Simulator/ processor type | General, Context aware | Schrödinger wavefunction | Falcon r4 T |
| Status | Active | Active | Active |
| Maximum shots | 20,000 | 20,000 | 20,000 |
| Maximum circuits | 300 | 300 | 100 |
| Basis gates | U1, U2, U3, U, P, R, RX, RY, RZ, ID, X, Y, Z, H, S, SDG, SX, T, TDG, SWAP, CX, CY, CZ, CSX, CP, CU1, CU2, CU3, RXX, RYY, RZZ, RZX, CCX, CSWAP, MCX, MCY, MCZ, MCSX, MCP, MCU1, MCU2, MCU3, MCRX, MCRY, MCRZ, MCR, MCSWAP, UNITARY, DIAGONAL, MULTIPLEXER, INITIALIZE, KRAUS, ROERROR, DELAY | | CX, ID, RZ, SX, X |
| Avg. CNOT Error | N/A | N/A | 1.032e-2 |
| Avg. Readout Error | N/A | N/A | 2.410e-2 |

instance. The classification model is trained on training data, and tested on test data. The regularization parameter C = [0.1, 10, 100, 1000] is used to control the trade-off between maximizing the margin of the decision boundary and minimizing the classification error on the training data.

### 5.8 Performance evaluation

The QSVM-based Classification model is evaluated using multiple performance metrics (expressed in Equation (4)) such as precision, accuracy, sensitivity, specificity, F1-score, and execution time.

$$\begin{aligned} \text{Precision} &= \frac{TP}{TP+FP} \\ \text{Accuracy} &= \frac{TP+TN}{TP+TN+FP+FN} \\ \text{Sensitivity (TPR)} &= \frac{TP}{TP+FN} \\ \text{Specificity} &= \frac{TN}{TN+FP} \\ F1-\text{Score} &= \frac{2(\text{Precision}\times\text{Sensitivity})}{\text{Precision}+\text{Sensitivity}} \end{aligned} \tag{4}$$

FP, TP, FN, and TN, in classification models, signify false positive, true positive, false negative, and true negative, respectively. The QSVM-based classification model is compared with the SVM-based model and their corresponding $2 \times 2$ confusion matrices along with AUC-ROC curves are presented. The confusion matrix compares the actual and predicted class labels in the QSVM-based model and shows the number of true positives, true negatives, false positives, and false negatives. The AUC-ROC curve is a graphical depiction of the performance of a binary classification model, plotting the true positive rate (TPR) against the false positive rate (FPR) at different classification thresholds. TPR is nothing but sensitivity and FPR is computed as $FPR = FP/(FP+TN)$. The efficiency of a classification model is evaluated by calculating the area under the ROC curve (AUC) from (0,0) to (1,1). The AUC value closer to 1 indicates that the classification model is efficient.

## 6. EXPERIMENTAL RESULTS

The proposed model was programmed in Python 3.6.9 using the Jupyter Notebook integrated with QISKIT, QML, and machine learning libraries. The kernel-based QSVM model for brain tumour classification is proposed and compared with variational QSVM- and SVM-based classification models. The dataset, comprising 3762 observations, underwent various pre-processing steps and EDA and split into 70% train data (2633 samples), and 30% test data (1129 samples). Figures 4 and 5 depict the relationship between some first-order and second-order features of the brain tumour dataset after applying PCA with the help of pair plots and heatmap, respectively. Heatmap and pair plots are generally used to see the correlation between the features.

The QSVM-based classification model proposed for the brain tumour dataset when run on *statevector_simulator*, classifies 92.6% of instances *i.e.* 460 out of 497 as true positives and 97.8% of instances *i.e.* 617 out of 632 instances as true negatives. Nonetheless, the model did misclassify 15 instances (2.4%) as false positives and 37 instances (7.4%) as false negatives. The kernel-based and variational QSVM both deliver almost the same results when run on the *statevector_simulator*. The classification model based on SVM produced 92.2% true positive results, accurately classifying 458 out of 497 instances, and 94.9% true negative results, correctly identifying 600 out of 632 instances when run on a local machine. However, the model also had 32 false positives (5.1%) and 39 false negatives (7.8%), which can be used to evaluate the model's performance and quality. The confusion matrices corresponding to the kernel-based QSVM model and SVM-based classification model are revealed in Figure 6.

Table 2 reveals the outcome of the proposed kernel-based QSVM model and its comparison with variational QSVM and SVM-based classification models for

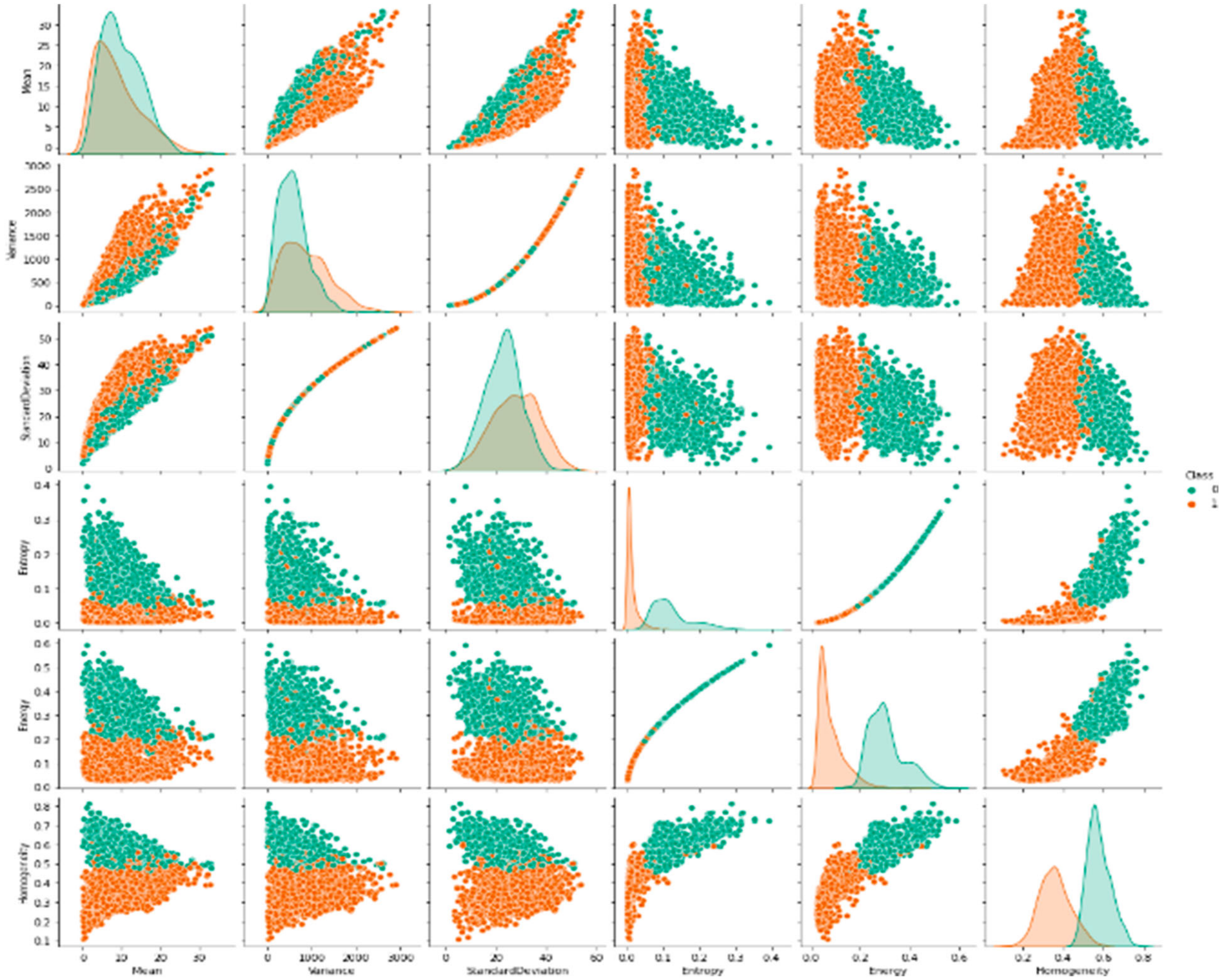



**Figure 4:** Visualization of hidden information inside the first- and second-order features of the brain tumour dataset

**Table 2: Performance analysis of classification models**

| Algorithm | Backend | Precision (%) | Sensitivity (%) | Specificity (%) | F1-score (%) | Accuracy (%) | Execution Time (s) |
|---|---|---|---|---|---|---|---|
| Kernel-based QSVM | *statevector_ simulator* | 95.5 | 95.5 | 95.5 | 95.5 | 95 | 0.491 |
| | *ibmq_lima* | 95.5 | 95 | 95.5 | 95.5 | 95 | 69.97 |
| | *ibmq_qasm_simulator* | 95.5 | 95.5 | 95 | 95.5 | 95 | 0.508 |
| Variational QSVM | *statevector_ simulator* | 95.5 | 95.5 | 95 | 95.5 | 95 | 19.15 |
| | *Ibmq_lima* | 93.33 | 93.33 | 93 | 93.33 | 93.33 | 124.98 |
| | *ibmq_qasm_simulator* | 93.33 | 93.33 | 93 | 93.33 | 93.33 | 27.03 |
| SVM | Local CPU environment | 93.33 | 93.5 | 93.5 | 93.33 | 93.5 | 92.30 |

the brain tumour dataset. When run on the *statevector_simulator*, the kernel-based and variational QSVM model deliver the same results but with high accuracy, sensitivity, precision, specificity, and F1-score, while also taking less time than the SVM-based model. The QSVM-based classification models demonstrated an overall accuracy of 95% on the *statevector_simulator*, which is 1.60% greater than the SVM-based classification model. The AUC-ROC curves of kernel-based QSVM and SVM-based classification models are shown in Figure 7. The QSVM-based classification model has an area under the ROC curve of 95%, whereas the SVM-based model has 93.5%. This indicates that the QSVM-based classification model performs better at distinguishing between classes than the SVM-based classification model.

The execution time for a kernel-based QSVM classification model when implemented using *statevector_simulator* on the local machine is 0.491s which is 188 times lesser than the execution time for a classical SVM-based classification model which is run on a local CPU environment. The *ibmq_qasm_simulator* takes 182 times lesser time to run a kernel-based QSVM classification model compared to the classical SVM-based classification model which is run on a local machine environment. The execution time for the kernel-based QSVM

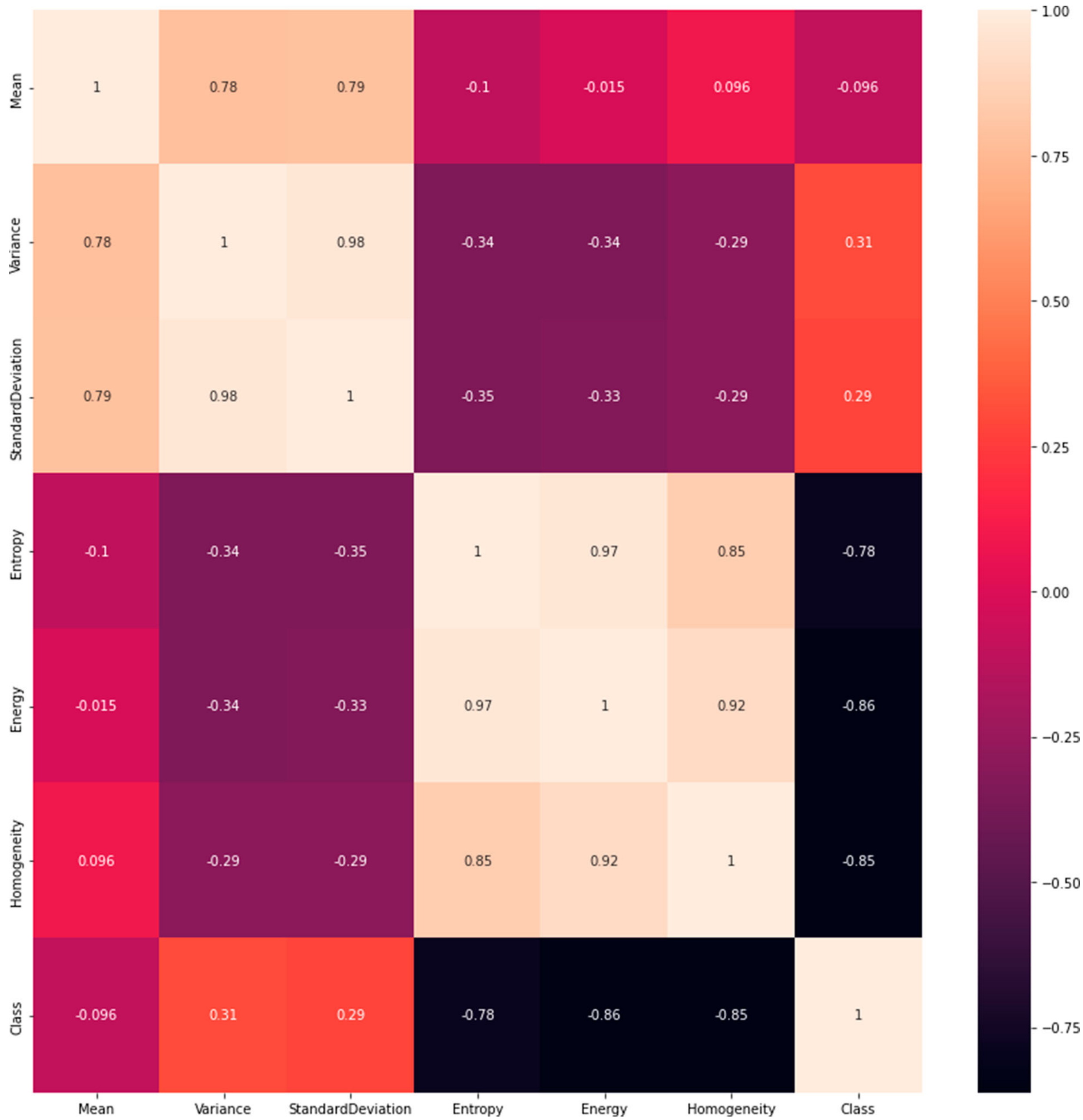
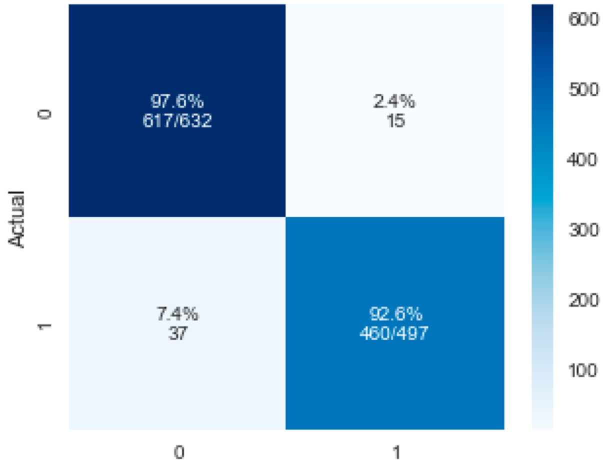


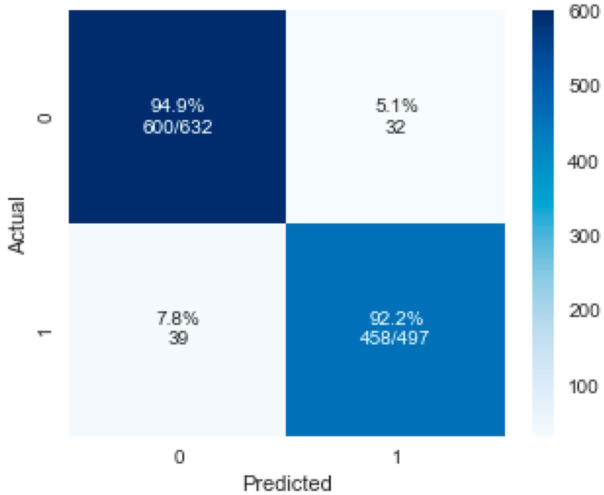

**Figure 5:** Visualization of correlation between first- and second-order features of the brain tumour dataset

classification model when implemented on ibmq_lima backend is 24.19% lesser than the execution time for classical SVM-based classification which runs on a local CPU environment. The superposition property enables quantum simulators/computers and algorithms to process multiple states at once, providing a quantum advantage in terms of speedup.

The variational QSVM-based classification model when run on the *ibmq_qasm_simulator* takes 70.7% lesser time than that of the classical SVM-based classification model which runs on the local CPU environment. In the case of *statevector_simulator*, it takes 79.25% lesser time to run a variational QSVM-based classification model for brain tumour compared to a classical SVM-based classification model which uses a local machine for processing. The 5-qubit *ibmq_lima* backend takes 35.40% more time to run the variational QSVM model compared to the classical SVM-based model which utilizes a local machine for processing. The reason behind that is the variational approach of QSVM requires two algorithms for the computation which increases the complexity and might lead to the increased execution time when it is implemented on the *ibmq_lima* backend.



(a)



(b)

**Figure 6:** Confusion matrix of (a) the kernel-based QSVM model (b) the SVM-based model for the brain tumour classification

## 7. CONCLUSION

The classical and quantum SVM-based classification models for the brain tumour dataset have been constructed and implemented practically on the various computational backends. Choosing the quantum feature map with an appropriate number of repetitions posed a challenge in this study, for efficient encoding of data points of brain tumour datasets into quantum states. ZZFeaturemap with 2 repetitions delivered more accurate results. It is observed that QSVM-based classification models, when implemented on quantum backends, offer a quantum advantage in terms of computational speed and accuracy. The accuracy improvement of 1.75% has been achieved by the kernel-based QSVM classification model implemented on all three backends compared to the classical SVM classification model implemented on a local machine. Similar results have been seen in the case of the variational QSVM-based classification model when implemented on *statevector_simulator*. The QML-based classification problem solved by using *statevector_simulator and ibmq_qasm_simulator* on a local machine was 188 times and 182 times faster than its classical equivalent. The 5-qubit real-time superconducting processor takes 24.19% less time to run a kernel-based QSVM model than the SVM-based model. It is anticipated from the results that quantum computers could efficiently solve classification or feature mapping problems for small datasets. In future, QML-based classification models can be developed by considering different quantum feature map combinations such as XZZ, YZZ, ZZZ, *XYZ, XZY, YZX, etc.* with an appropriate number of repetitions for feature mapping and also the QNN for the classification of brain tumours can be explored.

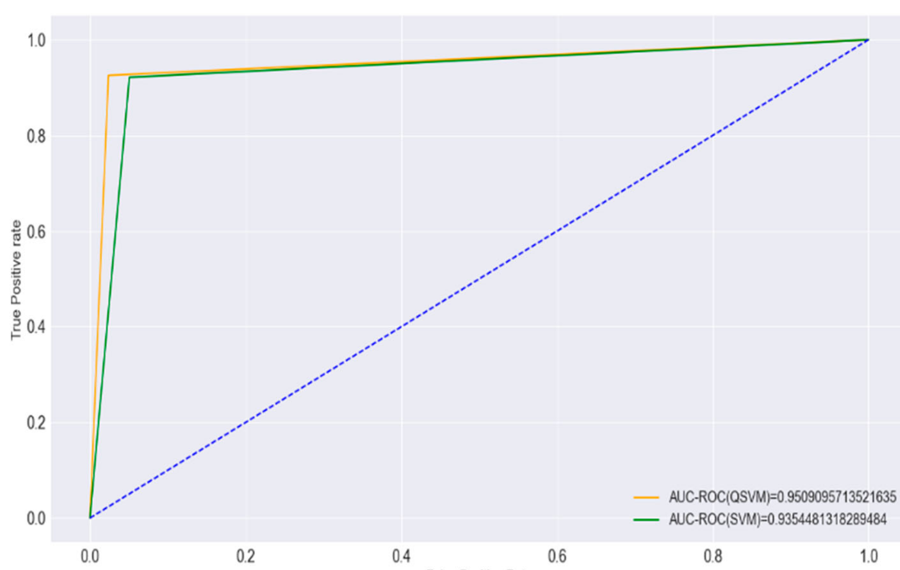



**Figure 7:** AUC-ROC curves of QSVM, and the SVM-based classification model for the brain tumour dataset

## ACKNOWLEDGEMENTS

The authors are highly grateful to IBM Quantum for providing access to their software and hardware computational resources through the cloud network.

## DISCLOSURE STATEMENT

No potential conflict of interest was reported by the author(s).

## ORCID

*Tarun Kumar* http://orcid.org/0000-0002-6295-9431

## REFERENCES

1. K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, London, 2012.
2. M. Schuld, I. Sinayskiy, and F. Petruccione, "An introduction to quantum machine learning," *Contemp. Phys*, Vol. 56, no. 2, pp. 172–85, 2015. DOI: 10.1080/00107514.2014.964942
3. C. W. Tsai, C. F. Lai, H. C. Chao, and A. V. Vasilakos, "Big data analytics: A survey," *J. Big Data*, Vol. 2, no. 21, pp. 1–32, 2015. DOI: 10.1186/s40537-015-0030-3.
4. D. Fisher, R. DeLine, M. Czerwinski, and S. Drucker, "Interactions with big data analytics," *Interactions*, Vol. 19, pp. 50–9, 2012. DOI: 10.1145/2168931.2168943.
5. E. Rieffel, and W. Polak, "An introduction to quantum computing for non-physicists," *ACM Comput. Surv*, Vol. 32, no. 3, pp. 300–35, 2000. DOI: 10.1145/367701.367709
6. M. A. Nielsen, and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge: Cambridge University Press, 2010.
7. M. Nagy, and S. G. Akl, "Quantum computation and quantum information," *Int. J. Parallel Emergent Distrib. Syst*, Vol. 21, no. 1, pp. 1–59, 2006. DOI: 10.1080/17445760500355678
8. J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, Vol. 549, no. 7671, pp. 195–02, 2017. DOI: 10.1038/nature23474
9. N. Mishra, *et al.*, "Quantum machine learning: A review and current status quantum machine learning," *Data Manag. Anal. Innov.*, Vol. 2, pp. 101–45, 2020.
10. P. Wittek. *Quantum Machine Learning*. Academic Press, 2014, pp. 73–83. DOI: 10.1016/B978-0-12-800953-6.00020-7
11. S. Al-Shoukry, T. H. Rassem, and N. M. Makbol, "Alzheimer's diseases detection by using deep learning algorithms: A minireview," *IEEE. Access.*, Vol. 8, pp. 77131–141, 2020. DOI: 10.1109/ACCESS.2020.2989396
12. H. Rajan, R. B. Ibrahim, and H. B. Hashim, "Automated brain tumor detection using machine learning: A bibliometric review," *World. Neurosurg.*, Vol. 175, pp. 57–68, 2023.
13. E. M. Senan, F. W. Alsaade, M. I. A. Al-Mashhadani, H. H. Theyazn, and M. H. Al-Adhaileh, "Classification of histopathological images for early detection of breast cancer using deep learning," *J. Appl. Eng. Sci*, Vol. 24, no. 3, pp. 323–29, 2021.
14. E. M. Senan, and M. E. Jadhav, "Analysis of dermoscopy images by using ABCD rule for early detection of skin cancer," *Glob. Transit. Proc.*, Vol. 2, no. 1, pp. 1–7, 2021. DOI: 10.1016/j.gltp.2021.01.001
15. J. Adcock, *et al*. "Advances in quantum machine learning." arXiv preprint arXiv:1512.02900, 2015.
16. Z. Ghahramani, "Probabilistic machine learning and artificial intelligence," *Nature*, Vol. 521, no. 7553, pp. 452–59, 2015. DOI: 10.1038/nature14541
17. L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *J. Artif. Intell. Res*, Vol. 4, pp. 237–85, 1996. DOI: 10.1613/jair.301
18. J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, Vol. 549, no. 7671, pp. 195–202, 2017. DOI: 10.1038/nature23474
19. J. M. Keller, M. R. Gray, and J. A. Givens, "A fuzzy k-nearest neighbor algorithm," *IEEE Trans. Syst. Man Cybern*, Vol. SMC-15, pp. 580–85, 1985. DOI: 10.1109/TSMC.1985.6313426
20. S. R. Safavian, and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Trans. Syst. Man Cybern*, Vol. 21, no. 3, pp. 660–74, 1991. DOI: 10.1109/21.97458
21. A. Ukil. *Intelligent Systems and Signal Processing in Power Engineering*. Berlin: Springer, 2007, pp. 161–26.
22. D. Berrar, "Bayes' theorem and naive Bayes classifier," *Encycl. Bioinform. Comput. Biol: ABC Bioinform*, Vol. 1, pp. 403–412, 2018.
23. S. Saini, P. K. Khosla, M. Kaur, and G. Singh, "Quantum driven machine learning," *Int. J. Theor. Phys*, Vol. 59, no. 12, pp. 4013–24, 2020. DOI: 10.1007/s10773-020-04656-1
24. M. Schuld, and N. Killoran, "Quantum machine learning in feature Hilbert spaces," *Phys. Rev. Lett*, Vol. 122, no. 4, pp. 040504, 2019. DOI: 10.1103/PhysRevLett.122.040504
25. K. P. Bennett, and E. J. Bredensteiner, "Duality and geometry in SVM classifiers," *ICML*, Vol. 2000, pp. 57–64, 2000.

26. H. Filali, J. Riffi, A. M. Mahraz, and H. Tairi, "Multiple face detection based on machine learning," in International Conference on Intelligent Systems and Computer Vision (ISCV) IEEE, 2018, pp. 1–8.

27. L. H. Thai, T. S. Hai, and N. T. Thuy, "Image classification using support vector machine and artificial neural network," *Int. J. Comput. Sci. Inf. Technol*, Vol. 4, no. 5, pp. 32–38, 2012.

28. P. Larranaga, *et al.*, "Machine learning in bioinformatics," *Brief. Bioinformatics*, Vol. 7, no. 1, pp. 86–112, 2006. DOI: 10.1093/bib/bbk007

29. S. M. Shamim, M. B. A. Miah, M. R. A. Sarker, and A. Al Jobair, "Handwritten digit recognition using machine learning algorithms," *Glob. J. Comput. Sci. Tech*, Vol. 18, pp. 17–23, 2018.

30. P. Rebentrost, M. Mohseni, and S. Lloyd, "Quantum support vector machine for big data classification," *Phys. Rev. Lett*, Vol. 113, no. 13, pp. 130503, 2014. DOI: 10.1103/PhysRevLett.113.130503

31. K. Shiba, K. Sakamoto, and T. Sogabe, "Variational quantum support vector machine based on Deutsch-Jozsa ranking," *Bull. Netw. Comput. Syst. Softw*, Vol. 9, no. 1, pp. 63–8, 2020.

32. S. Lloyd, M. Mohseni, and P. Rebentrost. "Quantum algorithms for supervised and unsupervised machine learning," arXiv preprint arXiv:1307.0411, 2013.

33. R. D. M. Simões, *et al.*, "Experimental evaluation of quantum machine learning algorithms," *IEEE. Access.*, Vol. 11, pp. 6197–6208, 2023.

34. S. S. Kavitha, and N. Kaulgud, "Quantum machine learning for support vector machine classification," *Evol. Intell*, Vol. 15, pp. 1–10, 2022.

35. G. Singh, M. Kaur, M. Singh, and Y. Kumar, "Implementation of quantum support vector machine algorithm using a benchmarking dataset," *Indian J. Pure Appl. Phys*, Vol. 60, no. 5, pp. 407–14, 2022.

36. J. Amin, *et al.*, "A new model for brain tumor detection using ensemble transfer learning and quantum variational classifier," *Comput. Intell. Neurosci*, Vol. 2022, pp. 1–13, 2022.

37. E. Akpınar, N. M. Duc, and B. Kesercİ, "The role of quantum-enhanced support vector machine using multiparametric MRI parameters in differentiating medulloblastoma from ependymoma," in IEEE International Conference on Quantum Computing and Engineering (QCE), 2022, pp. 882–85.

38. W. Ayadi, I. Charfi, W. Elhamzi, and M. Atri, "Brain tumour classification based on hybrid approach," *Vis. Comput.*, Vol. 38, pp. 107–17, 2022. DOI: 10.1007/s00371-020-02005-1.

39. N. S. Shaik, and T. K. Cherukuri, "Multi-level attention network: application to brain tumour classification," *SIViP*, Vol. 16, pp. 817–24, 2022. DOI: 10.1007/s11760-021-02022-0

40. R. Hao, K. Namdar, L. Liu, and F. Khalvati, "A transfer learning–based active learning framework for brain tumour classification," *Front. Artif. Intell*, Vol. 4, pp. 635766, 2021. DOI: 10.3389/frai.2021.635766

41. M. Mudda, R. Manjunath, and N. Krishnamurthy, "Brain tumour classification using enhanced statistical texture features," *IETE J. Res*, Vol. 68, no. 5, pp. 3695–06, 2022. DOI: 10.1080/03772063.2020.1775501

42. J. Bohaju. Brain tumour, Kaggle, 2020, DOI: 10.34740/KAGGLE/DSV/1370629.

43. L. C. Wang, "Experience of data analytics in EDA and test—principles, promises, and challenges," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst*, Vol. 36, no. 6, pp. 885–98, 2017. DOI: 10.1109/TCAD.2016.2621883

44. S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum principal component analysis," *Nat. Phys*, Vol. 10, no. 9, pp. 631–33, 2014. DOI: 10.1038/nphys3029

45. 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. 209–12, 2019. DOI: 10.1038/s41586-019-0980-2

46. M. M. Hossain, *et al.*, "Analyzing the effect of feature mapping techniques along with the circuit depth in quantum supervised learning by utilizing quantum support vector machine," in 2021 24th International Conference on Computer and Information Technology (ICCIT), IEEE, 2021, pp. 1–5.

47. M. Schuld, and F. Petruccione, "Quantum models as kernel methods," in *Machine Learning with Quantum Computers*, 2, R. Laflamme, *et al.*, Ed. Cham: Springer, 2021, pp. 217–45.

48. IBM Inc. IBM, Quantum information science kit, Available: https://qiskit.org/aqua.

49. IBM Inc. IBM Quantum services, Available: https://quantum-computing.ibm.com/services/resources?tab = simulators&system = simulator_statevector, Accessed March, 11, 2021.

50. IBM Inc. IBM Quantum services, Available: https://quantum-computing.ibm.com/services/resources?tab = simulators&system = ibmq_qasm_simulator, Accessed March, 11, 2021.

51. IBM Inc. IBM Quantum services, Available: https://quantum-computing.ibm.com/services/resources?tab = systems&skip = 20&system = ibmq_lima, Accessed March, 11, 2021.

## AUTHORS


**Tarun Kumar** received the BTech degree in electronics and communication engineering from Punjab Technical University, Jalandhar, India in 2013 and the ME degree in electronics and communication engineering from Panjab University, Chandigarh, India in 2018. He is currently pursuing PhD in electronics and communication engineering at the Department of Electronics and Communication, Sant Longowal Institute of Engineering and Technology (Deemed to be University) Longowal, Sangrur, Punjab, India. His areas of interest are quantum computing, quantum search algorithms and quantum machine learning.

**Corresponding author. Email:** tarunkumar2992@gmail.com

**Dilip Kumar** received the BE degree in electronics and telecommunication engineering from the Army Institute of Technology, University of Pune, India, in 2000, the ME degree in electronics product design and technology from the Panjab University, Chandigarh, India, in 2003, and PhD degree in faculty of engineering and technology from the MM University, Mullana, India, in 2010. He is currently working as a professor at the Department of Electronics and Communication Engineering, Sant Longowal Institute of Engineering and Technology (Deemed to be University) Longowal, Punjab, India. He appeared in the world's top 2% scientists list published by Stanford University in 2021. His research interests include quantum computing, quantum machine learning, embedded systems, electronic system design, wireless sensor networks, and vehicular ad hoc networks. He has published more than 60 research papers in reputed international journals and conferences.

**Email:** dilip.k78@gmail.com

**Gurmohan Singh** received the BTech degree in electronics and communication engineering from Giani Zail Singh College of Engineering and Technology Bathinda, India in 2001, the MTech degree in microelectronics from Panjab University, Chandigarh, India in 2005, and PhD degree in electronics and communication engineering from NIT Jalandhar, India in 2019. He is currently working as a joint director at the Cyber Security Technology Division (CSTD), Centre for Development of Advanced Computing (C-DAC), Mohali, India. His areas of interest are quantum computing, quantum machine learning, VLSI, computational nanoscience, molecular electronics and MEMS/NEMS sensors.

**Email:** gurmohan@cdac.in