Tomono and Natsubori *EPJ Quantum Technology* (2022) 9:35
https://doi.org/10.1140/epjqt/s40507-022-00157-8

EPJ Quantum Technology
a SpringerOpen Journal

RESEARCH Open Access

Performance of quantum kernel on initial learning process

Takao Tomono[1]* and Satoko Natsubori[1]

*Correspondence:
takao.tomono@toppan.co.jp
[1] Digital Innovation Div., Toppan Inc., 1-5-1 Taito, Taito, Tokyo 110-8560, Japan

Abstract

For many manufacturing companies, the production line is very important. In recent years, the number of small-quantity, high-mix products have been increasing, and the identification of good and defective products must be carried out efficiently. At that time, machine learning is a very important issue on shipping inspection using small amounts of data. Quantum machine learning is one of most exciting prospective applications of quantum technologies. SVM using kernel estimation is one of most popular methods for classifiers. Our purpose is to search quantum advantage on classifier to enable us to classifier in inspection test for small size datasets. In this study, we made clear the difference between classical and quantum kernel learning in initial state and propose analysis of learning process by plotting ROC space. To meet the purpose, we investigated the effect of each feature map compared to classical one, using evaluation index. The simulation results show that the learning model construction process between quantum and classical kernel learning is different in initial state. Moreover, the result indicates that the learning model of quantum kernel is the method to decrease the false positive rate (FPR) from high FPR, keeping high true positive rates on several datasets. We demonstrate that learning process on quantum kernel is different from classical one in initial state and plotting to ROC space graph is effective when we analyse the learning model process.

1 Introduction

Quantum machine learning (QML) is one of most exciting prospective applications of quantum technologies [1–5]. Kernel estimation is one of the methods to estimate the whole distribution from a finite number of sample points and a typical example of non-parametric estimation that cannot be expressed by parametric estimation. The inner product space is used for discrimination. Therefore, kernel estimation matches the mapping to the Hilbert space, and it is promising method for SVM as classifier.

Support vector machine (SVM) is the most often used method in various machine learning [6–9]. This method is based on statistical machine learning, which allows the construction of training models with relatively little data. In recent years, kernel estimation SVM has been widely used as one best method [10–12]. Kernel SVM is widely used for pattern recognition and other imaging applications, as we can separate non-linear feature spaces by using inner products.

For many manufacturing companies, the production line is very important. In recent years, the number of small-quantity, high-mix products have been increasing, and the classification of good and defective products must be carried out efficiently. The classification includes image data, text, and sound. Image classification is widely used in remote sensing [6], biological inspection [13–15], building and civil engineering [16] and manufacturing [17–19]. The inspection of defective products is a very important issue in the inspection process in the manufacturer. The learning model of two-class classification is used in such inspection processes. Recently, we have limited training size (good and defective products), as many products have been produced in small quantities and in many varieties. Therefore, we need a machine learning model that enables limited and small data for classification.

However, we have two issues with the kernel estimation. One is calculation cost. We need a huge calculation cost as the embedding function into the feature space increases dramatically when the feature volume increases. The other is the limitation of the embedding function. We must treat a complicated function when we use the kernel trick in SVM. As a means of solving the above problems, there are two attempts to use kernel estimation to embed feature maps with quantum entanglement in the Hilbert space. One is the quantum kernel SVM, which introduced Z-ZZ feature maps as quantum entanglement in an exponentially large feature space [20]. The other is the kernel estimation neural network, and they propose methodology for assessing potential quantum advantage in learning tasks [21].

Our purpose is to obtain a highly accurate learning model that classifies small training size imaging data in shipping inspection.

In this work, we investigated the difference between classic and quantum learning process. In the Sect. 2, we describe relative work using quantum kernel estimation. In the Sect. 3, we explain preparation of dataset and quantum circuits we used in this work. In the Sect. 4, we denote simulation results by using quantum simulator and actual machine. First, we look into an effect of entanglement using Pauli-, Pauli-ZZ feature map compared to classic kernel estimation as Sect. 4.1. Second, we check learning process using accuracy and F1-score as evaluation index as Sect. 4.2. Thirdly, we propose plotting onto ROC space graph using confusion matrix as Sect. 4.3. Fourthly, we described first trial using our product as Sect. 4.4. Generally evaluation index: AUC is used in ROC graph [22–25]. Here, we use new plotting method that is different from method used conventional ROC graph. In the Sect. 5, we discuss the meaning of plotting onto ROC space graph. In the Sect. 6, we conclude our work and describe future outlook.

2 Related work

We described two related works with quantum kernel estimation. One is kernel estimation SVM, and the other is kernel estimation neural network.

Two quantum algorithms on a 5-qubits superconducting processor are proposed and implemented experimentally to solve cost issues described above [20] in 2019. To do speed up for cost problem, they though utilization of an exponentially large quantum state space through controllable entanglement and interference.

One method is to implement the quantum variational classier builds as variational quantum circuits on the processor [26, 27] and the other method is to estimate the kernel function and optimize the classier directly by using quantum kernel estimator [28]. They proposed Z-ZZ feature map as the kernel function in the quantum circuits. This feature map use combination of Pauli-Z feature map and ZZ feature as quantum entanglement.

A methodology for assessing potential quantum advantage in learning tasks was developed [21] in 2021. They referred that classical machine learning models can be competitive with quantum models with the help of data even if they are tailored to quantum problems. The scheme is explained by the cartoon of the geometry (kernel function) defined by classical and quantum ML models.

They propose a projected quantum model as shown in the cartoon that provides a simple and rigorous quantum speed-up for a learning problem in the fault-tolerant regime. For near-term implementations, they use 30-qubits actual gate-based quantum computer for demonstrating quantum advantage.

We focus feature map with/without entanglement on quantum kernel circuit learning, with reference to above research.

3 Preparation of datasets and circuits

The conventional datasets we used are Iris, heart disease, and wine. The summary of each dataset is shown in Table 1. Heart disease is a two-class classification dataset with attributes of 13. Wine is a three-class classification dataset with attributes of 13. Iris is a three-class classification dataset with attributes of 4. We create two class datasets Iris_2 with attributes of 4 form original Iris dataset (we call it Iris_3), which consist of versicolor and Virginia. Using these data, we can compare two-class classification with three-class classification with attributes of 4 and 13.

Figure 1 shows quantum circuits diagram for quantum kernel SVM. Fig. (a), (b), (c) and (d) stand for quantum circuits diagram, detailed quantum circuits diagram using Y Pauli-feature map, Z Pauli-feature map and Y-ZZ feature map. Y-ZZ means quantum entanglement feature map as described later. Here, we use classical and quantum hybrid systems. We perform training and prediction on the classical SVM by using the gram matrix calculated on the quantum circuit. The distance between the classical data x and x' is calculated by the kernel $\kappa(x, x')$. By means of a nonlinear mapping $\varphi(x)$ embedding the data into the quantum feature space, it can be expressed in the feature space as follows.

$$\kappa\left(x, x^2\right) = \left|\left\langle\varphi\left(x'\right) \mid \varphi(x)\right\rangle\right|^2 \tag{1}$$

We prepare $\varphi(x) = S(0)|0\rangle$ as a data encoding from classical to quantum data, first. To obtain the inner product $\kappa(x, x')$, we prepare $|\varphi(x)\rangle = S(x')^\dagger S(x)|0\rangle$ as the initial state of the quantum circuit. The probability of measuring on $|0\rangle$ for all qubits is as follows.

$$\left\langle 0 \mid S^\dagger(x) S\left(x'\right) \mid 0\right\rangle\left\langle 0 \mid S^\dagger\left(x'\right) S(x) \mid 0\right\rangle = \left\langle\varphi(x) \mid \varphi\left(x'\right)\right\rangle\left\langle\varphi\left(x'\right) \mid \varphi(x)\right\rangle \tag{2}$$

Table 1 Summary of datasets. Iris_3 stands for original Iris datasets, and Iris_2 stands for created Iris datasets except for setosa. Attributes stand for number of attributes, that is features. Sample/class stand for Number of samples in each class

Dataset	Classes (Attributes)	Sample/class	Remark
Heart disease	2 (13)	138, 165	Positive, Nagative
Wine	3 (13)	59, 71, 48	Three different cultivars
Iris_2	2 (4)	50, 50	Versicolor, Versinica
Iris_3	3 (4)	50, 50, 50	Versicolor, Versinica, Setosa

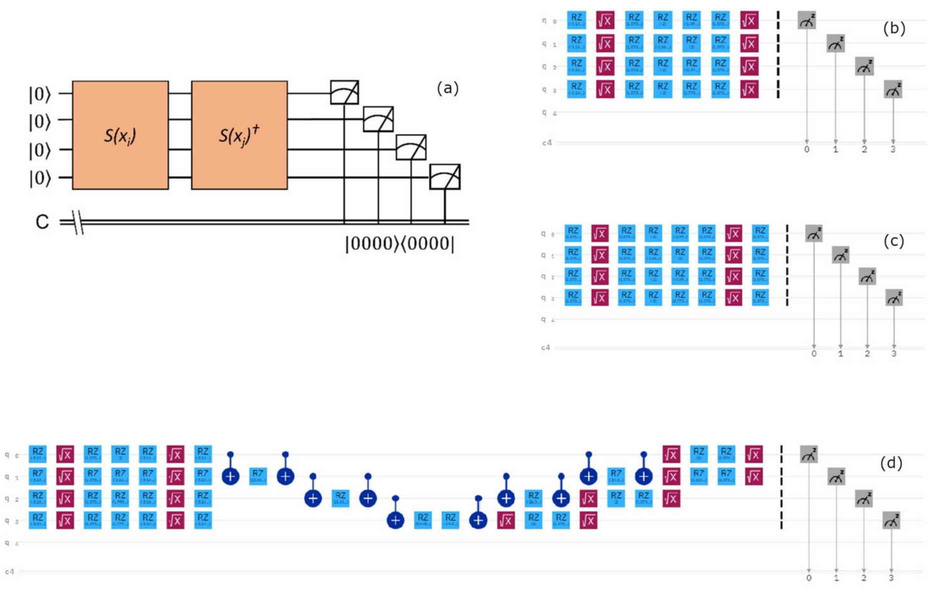

Figure 1 Quantum circuits diagram for quantum kernel SVM. Quantum circuits for quantum kernel SVM are denoted in the case of attributes (feature volume) of 4. The quantum data is created by these circuits illustrated above. After measurement, these quantum data are inserted into classical machine learning. Fig (**a**) denote quantum circuits diagram. Fig (**b**), (**c**) and (**d**) denote detailed circuit diagram using y-, z-, and y-zz-feature map on IBM qiskit simulator

Here $S(x)$ is the inner product between the quantum encoded data using quantum kernel estimation. Each feature map is then embedded into the inner product to optimize the parameters. The matrix component of the entire Gram matrix is obtained from a combination of the inner product. The parameters of the kernel estimation are optimized using rotation gates with/without entanglement in Eq. (2). We use rbf as classical kernel of SVM in this work.

Gate-based quantum simulator and computer were used. The simulation was performed by using IBM qiskit and confirmed by blueqat. Actual 5-qubits quantum computer used in this work was ibmq Bogota. The number of shots is 1024, the number of seed is 10,598.

We checked the testing accuracy (accuracy) and F1-score as evaluation indices when the ratio of training size was changed. Here, training size started from 6 for Iris_2, 9 for Iris_3, 8 for heart disease, and 9 for wine. As mapping in the Hilbert space, Pauli-X, -Y, and -Z feature maps and X-ZZ, Y-ZZ and Z-ZZ feature maps with entanglement were used.

4 Results

4.1 Effect of entanglement

To compare quantum kernel with/without entanglement, we embedded each feature map. Figure 2 shows the effect of each feature map on the quantum kernel SVM (qkSVM) on the heart disease and wine datasets. Here, we show a comparison between the classical kernel SVM (ckSVM) and qkSVM embedded with Y, Z, X-ZZ, Y-ZZ, and Z-ZZ feature maps (qkSVM with Y, Z, X-ZZ, Y-ZZ, and Z-ZZ) on heart disease and wine datasets with attributes of 13.

As the training size become larger, the accuracy increases. These accuracies become more than 0.8 at a training size of 72 for ckSVM, qkSVM with Y and Z. For all datasets,

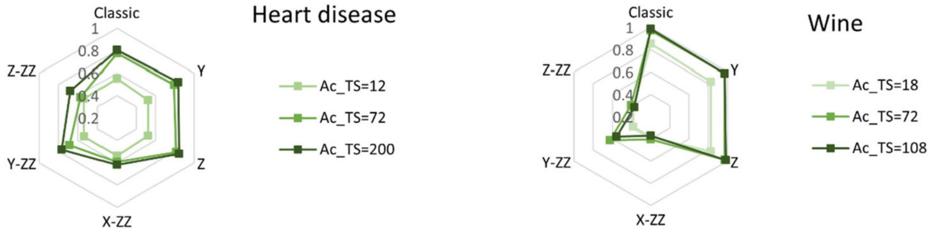

Figure 2 Effect of each feature map on training accuracy on quantum kernel support vector machine (qkSVM). Here we used Heart disease and Wine datasets. Here, Ac and TS stand for accuracy and Training Size. Classic, Y, Z, X-ZZ, Y-ZZ, and Z-ZZ stand for each feature maps. After measurement, the data are inserted into classical machine learning

the accuracy of qkSVM with X-ZZ and Z-ZZ with entanglement was lower than that of qkSVM with Y-ZZ.

When the training size for heart disease was 200, the accuracy of qkSVM with Y and Z was 0.835 and 0.845, that of qkSVM with Y-ZZ was 0.767, and that of ckSVM was 0.806. The values of qkSVM with X-ZZ and Z-ZZ were 0.621 and 0.680, respectively. When the training size of Wine was 108, that of ckSVM, qkSVM with Z and Y were 0.986, 0.971 and 0.986. These accuracy data are almost the same values, which are approximately double the accuracy of qkSVM with X-ZZ, Y-ZZ and Z-ZZ (0.371, 0.557 and 0.371).

From the above, introducing quantum entanglement is not effective on the accuracy when we use datasets of Iris, Heart diseases and Wine in this work. On the other hand, the qkSVM with Y and Z would have the same or better performance than ckSVM. Moreover, the outline of the learning model is thought to build on the range of less than 72.

4.2 Learning process

The confusion matrix is important indicator on the classification problem. Accuracy is an indicator of how correct the prediction was. Precision is an index to see how correct what was predicted to be positive. Recall is an index to see how many of the actual positive results could be predicted to be positive. The F1-score is the harmonic mean of Precision and Recall. To analyze the process on the learning model, we had better compare both accuracy with F1-score.

Figure 3 shows the relationship between the training size and index on each dataset. Here, the training size is less than 100. In the Iris_2, Iris_3 and Wine datasets, the evaluation index (Accuracy and F1-score) rises dramatically when each training size is less than 20. Moreover, machine learning model using qkSVM shows higher accuracy and F1-score than that using ckSVM except for Heart disease. The values of accuracy and F1-score were almost the same as when the training size was 20 or more. The difference between accuracy and F1-score of qkSVM is in the order of heart disease > Iris_2 > Wine.

We can use within 5 qubits an actual quantum computer (ibmq bogota). Under this limitation, we can calculate the classification of attributes (feature volume) of 4. Experiments were carried out on Iris_2 and Iris_3. The shot number is 1024, and we used the average value on 10 times. The index value performed on the actual quantum computer machine almost coincides with the locus of the simulator.

In the 2-class classification, the accuracy and F1-score of qkSVM on the Iris_2 datasets were almost the same when the training size was 20 or more. When the training size was

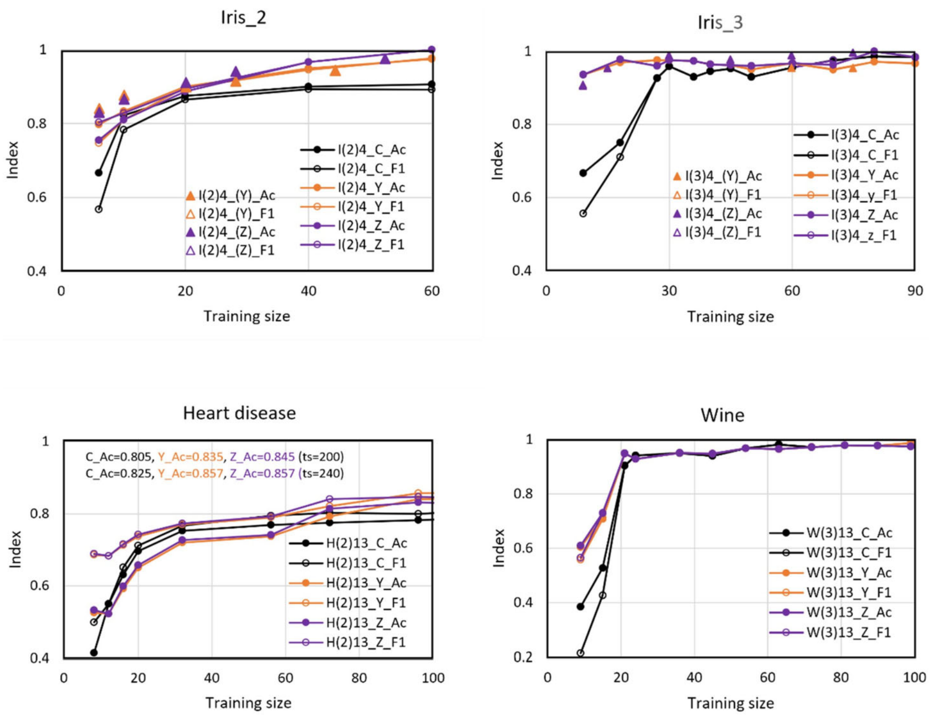

Figure 3 Relationship between training size and Index. Here, each filled circle is Accuracy and open circle is F1-score obtained by quantum simulation. Triangle is actual quantum machine (ibmq bogota). C, Y and Z is classic kernel learning and quantum kernel learning with Y and Z-feature map. I(2)4_C_Ac means accuracy classical kernel learning on 2-class-classification and attributes of 4

60, the accuracy and F1-score of qkSVM with Z became 1.000. In the datasets of heart disease, the order of index was F1-score of qkSVM > accuracy and F1-score of ckSVM > accuracy of qkSVM when the training size was less than 60. Accuracy and F1-score on quantum kernel keep large value compared to classic kernel on training size = 240 (training size: testing size = 08: 0.2). However, the order of the index was accuracy and F1-score of qkSVM > accuracy and F1-score of ckSVM when the training size was 60 or more.

In the 3-class classification, the accuracy and F1-score of qkSVM on the Iris_3 datasets showed a higher value than those of ckSVM when the training size was less than 60. The difference in the index between qkSVM and ckSVM decreased when the training size exceeded 60. In the datasets of Wine, qkSVM shows higher accuracy and F1-score than ckSVM when the training size is less than 20. However, each index of qkSVM and ckSVM become almost the same when the training size exceeds 20.

Table 2 shows the training size when the accuracy and F1-score reached 1.000 and the index value when the training size was 80% of the total data size in the case that these indexes did not reach 1.000. When the accuracy and F1-score reach up to 1.000, the training size on qkSVM is less than that on ckSVM. In the case of heart disease, these indices do not reach 1.000, and these indices of qkSVM are higher than those of ckSVM. The reason is why the attributes (feature volume) of the heart disease dataset is larger than that of the Iris_2 dataset and the heart disease dataset of the actual problem is complex compared to the wine dataset of the toy problem.

Table 2 Training size when accuracy and F1-socre are reached up to 1.000 and index value when training size is 80% of total data size in the case that these index does not reach 1.000. Ac, F1, and ts represent the accuracy, F1-score, and training size, respectively

Kernel	Heart disease	Iris_2	Wine	Iris_3
Classic	Ac:0.825, F1:0.825 (ts = 240)	Ac:0.750, F1:0.667 (ts = 80)	Ac:0.971, F1:0.971 (ts = 144)	Ac:1.000, F1:1.000 (ts = 110)
Y	Ac:0.857, F1:0.857 (ts = 240)	Ac:1.000, F1:1.000 (ts = 75)	Ac:1.000, F1:1.000 (ts = 144)	Ac:1.000, F1:1.000 (ts = 110)
Z	Ac:0.857, F1:0.825 (ts = 240)	Ac:1.000, F1:1.000 (ts = 60)	Ac:1.000, F1:1.000 (ts = 144)	Ac:1.000, F1:1.000 (ts = 80)

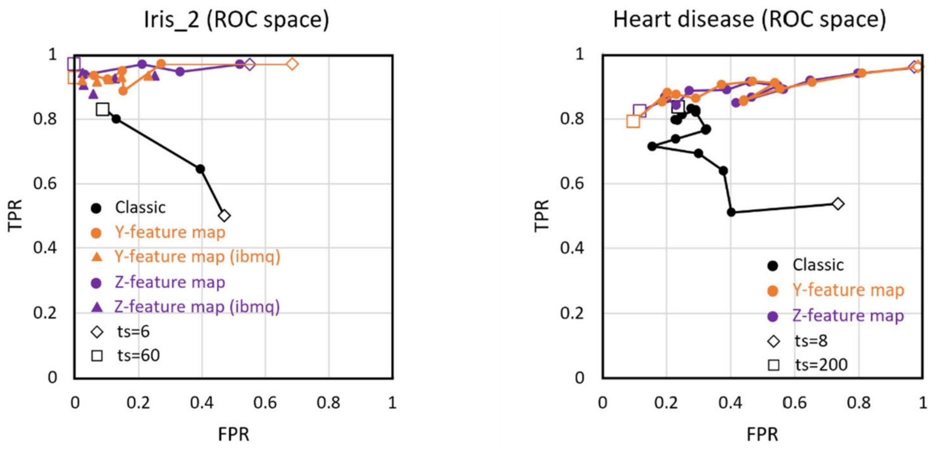

Figure 4 The plotting onto ROC space. FPR and TPR show false positive rate and True positive rate. FPR is FP/(FP + TN) on confusion matrix. TPR is TP/(TP + FN), which is the same as Recall. Open diamond and square shape stand for training size of 8 and 200 on heart disease dataset, 6 and 60 on Iris_2 dataset

From the above, the outline of the learning model is formed up to approximately 20 data points. After that, each parameter on the learning model is finely regulated by the Pauli-feature map when the training size increases.

We found that the learning model on qkSVM is constructed by using a smaller training size than that on ckSVM in the initial state. Moreover, we confirmed that the value of accuracy and F1-score on quantum kernel is larger than these on class kernel.

4.3 Model construction process

To clarify the learning process in the initial state, we used an ROC space graph that is different from conventional ROC curve. ROC curve was used to analyze the learning model construction on classical machine learning by using the true positive rate (TPR) and false positive rate (FPR). The TPR and FPR are obtained from the confusion matrix.

Figure 4 shows the plotting onto ROC space. First, we observed plotting FPR and TPR on heart disease onto the ROC space. In the case of ckSVM, the plotted point moves from FPR = 0.73 and TPR = 0.54 on a training size of 8 to FPR = 0.23 and TPR = 0.83 on a training size of 200. In the case of qkSVM with Y and Z, the plotted point moves from FPR = 0.99 and TPR = 0.96 on a training size of 8 to FPR = 0.1 and TPR = 0.8 on a training size of 200. Here, although not shown in this Fig. 4, we will consider the case where AUC is 0.85. The plotted point of ckSVM on ts = 200 was below the curve. That is to say, the

Table 3 Training accuracy v.s.testing accuracy on heart dataset. Accuracy indicates testing accuracy. Accuracy indicates test accuracy. At 12 trainings, it can be considered that learning has just started, and at 72,120 trainings, it can be considered that model building is in the final stage

Training size	Classic		Y feature map		Z feature map	
	Training Accuracy	Testing Accuracy	Training Accuracy	Testing Accuracy	Training Accuracy	Testing Accuracy
12	0.667	0.550	1.000	0.522	1.000	0.523
72	0.889	0.775	0.986	0.792	0.986	0.814
200	0.905	0.806	0.980	0.835	0.980	0.845

AUC is less than 0.85. On the other hand, the position of qkSVM on ts = 200 was on the top of the curve. That is to say, the AUC on qkSVM is greater than 0.85.

We observed a similar trend for Iris_2. In the case of ckSVM, the plotted point moves from FPR = 0.2 and TPR = 0.5 on a training size of 6 to FPR = 0.05 and TPR = 0.83 on a training size of 60. In the case of qkSVM with Y, the plotted point moves from FPR = 0.27, TPR = 1 on a training size of 6 to FPR = 0, TPR = 0.95 on a training size of 60. In the qkSVM with Z, the plotted point moves from FPR = 0.49, TPR = 1 to FPR = 0, TPR = 1 on a training size of 60. When we use the actual quantum computer, the plotted point moves from FPR = 0.23, TPR = 0.91 (FPR = 0.25, TPR = 0.91) on a training size of 6 to FPR = 0.07, TPR = 0.9 (FPR = 0.06, TPR = 0.85) on a training size of 52 in the qkSVM with Y (Z). The actual data show similar trends with the simulator, although the TPR on the actual quantum computer was smaller than that on the simulator.

We measured the training accuracy in addition to the testing accuracy on heart disease. The results are shown in Table 3. The training sizes are 12, 72, and 120, as the same settings used in the laser chart in Fig. 2. The training accuracy of ckSVM increases gradually as training size become larger, as well as testing accuracy. In other words, the learning model is gradually constructed as the number become larger. On the other hand, quantum kernel learning using qkSVM with Y and Z feature maps showed high training accuracy around 1 on the training size of 12, 72 and 200. High training accuracy was maintained even if the training size become larger. This trend was different from the testing accuracy. We found that this result indicates that the construction of the learning model is completed with the data area used for training in the case of qkSVM.

As described above, we found that the learning process on ckSVM is different from that on qkSVM. The learning process on qkSVM always maintains a higher TPR, and the learning model starts with a high TPR and a high FPR in the initial states. Keeping the TPR of almost 1 while decreasing the FPR, a learning model on qkSVM is constructed.

4.4 First trial on actual products

So far, we have run simulations on a balanced toy dataset. Real products include imbalanced datasets. Table 4 shows the results of testing accuracy applied to defect detection of industrial products in our factories.

Although the number of original image data exceeds 10,000, the defective product rate is less than 1%. Of these, 400 (good product of 300 and defective product of 100) were extracted. Image processing was performed as preprocessing for machine learning. Then, we selected 10 features by using principal component analysis (PCA) of classical machine learning. Here, cumulative contribution of the PCA is greater than 80% when the attributes (feature volume) is 10. After that, we performed classification with ckSVM and qkSVM

Table 4 Testing accuracy applied to defect detection of industrial product. Prod. and (Q)ML stand for industrial product and (quantum) machine learning. Defective product is less than 1% on developing products. ⇒ means preprocessing. After preprocessing, we select 10 features by principal component analysis (PCA). C, Z stand for ckSVM and qkSVM using Z feature map

	Original data		Imbalance dataset for (Q) ML					
	Quantity(defect)	⇒	Size	Good: Defect	Training	Testing	Accuracy: C	Accuracy: Z
Prod.	>10,000 (<100)		160	3:1	40	120	0.93	0.97

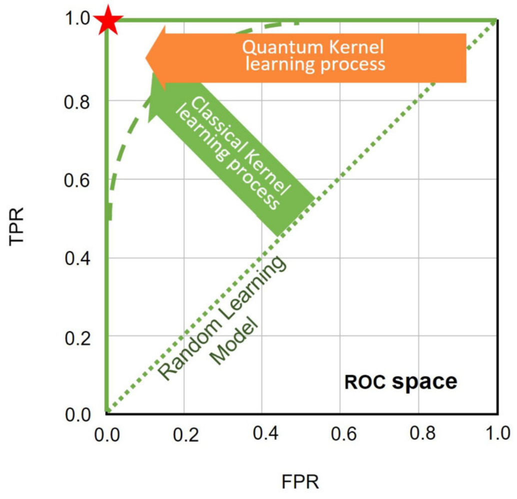

Figure 5 Hypothesis: Quantum & classical Learning Process on ROC space. Dotted, dashed and continuous green line stand for random learning model (AUC = 0.5), acceptable learning model (AUC > 0.8) and ideal model

with training size of 40 and testing size of 120. The ratio of good and defective product is 3:1. Then we got the testing accuracy. The accuracy of ckSVM and qkSVM with Z is 0.93 and 0.97. We obtained a little bit higher accuracy of qkSVM than one of ckSVM. Although not shown in Table 4, we confirmed that the plots of (FPR, TPR) almost lies on the classical and quantum trajectories, respectively as shown in Fig. 4.

We have just started a trial of detecting defective product in factory products. From now on, we would like to accumulate data and obtain reliable knowledge.

5 Discussion

The calculations on quantum computers are characterized by quantum operations in quantum circuits. The quantum operation get started with encoding from classical data to quantum data. Encoding is performed by projection onto the Hilbert space. This projection stands for mapping. The encoding from classical data to quantum data corresponds to $\varphi(x) = S(0)|0\rangle$ as shown in Eq. (1).

Figure 5 shows Hypothesis: Quantum & classical Learning Process on ROC space. The dotted green line stands for random learning model. As the learning make progresses, AUC curve become dashed green line when AUC exceeds over 0.8. And ideal learning model become continuous green solid line. When we calculate FPR and TPR using each TP, FN, FP and TN, random learning model is at FPR = 0.5 and TPR = 0.5 and ideal learning model is red filled star sign. The results obtained on Sect. 4 means that the true positive rate become large and the false positive rate decrease (green arrow in the figure) as the learning progress. Therefore, our result of classical simulation is reasonable.

We observed that the learning process got started with a high TPR and FPR as quantum kernels are used. Keeping on high TPR, the FPR become small as learning progress (orange arrow in the figure). We are thinking as follows. Classical data are transformed into quantum data and embedded in a Hilbert space. Then, the learning model is completed within the randomly set training data range as can be inferred from the results in Table 3. At initial learning, the training data is randomly and sparsely scattered. Therefore, the model building on quantum kernel learning can be considered to have a wide tolerance. As a result, the model is likely to exhibit high TPR and FPR in the initial learning process. Then, as training progresses, the density of the data space increases, so the learning model is expected to be less tolerant and have a lower FPR.

Let's think about quantum operation based on a maze. There are various routes in the maze. When entering the entrance, all routes including the route to the correct exit are listed as candidates at the same time. This is considered to correspond to superposition. After that, each route is bifurcated, and interfered. As the result, the routes that are not correct are weakened, and the routes that are correct are strengthened, which become the highest probability. After that, the quantum state collapses in the measurement. The above description is an image of quantum calculation.

We are thinking that the fact that FPR and FPR start near 1 means that all cases are candidates at the same time, so learning starts from the superposition state.

6 Conclusion and outlook

We investigate the difference between classic and quantum kernel learning by using several evaluation indices. The simulation results got several suggestions. In general, they say that quantum entanglement contributes to improved accuracy in classification as we can embed more future map in the Hilbert space. However, we could not show the effect of quantum entanglement on several datasets we selected. Our results suggested that the quantum learning model building process is different from classical one on several datasets we selected. From these results, we plotted onto ROC space according to training size. As the result, we could recognize the difference between classical and quantum kernel learning in initial state. Therefore, we propose utilization of ROC space graph to investigate initial learning process. We made clear the initial behavior of quantum learning process from ROC space graph.

Recall rate (TPR) is that we want to avoid the risk of erroneously predicting a defective product (Positive) as a good product (Negative), and to classify cases where there is a suspicion of a defective product (Positive) without omission. High TPR is important in such a case. Plotting in the ROC space is also important from such a point of view. Our purpose is to efficiently detect defective products for high-mix, low-volume production. We believe that it can be an evaluation tool suitable for this purpose.

Quantum computers and classical computers are expected to coexist in the future. It is important to distinguish between computations that quantum computers are good at and computations that classical computers are good at. When we look into implementation of quantum machine learning to factory, it is necessary to select calculations that quantum computers are good at. To do so, we need to accumulate results calculated by quantum computers to determine which computations they are good at. From now on, we would like to build a useful classifier through trials.

Acknowledgements
We would like to thank IBM for free of charge 5-qubit quantum computer used for this study and our team member (Hadamard Team) on our company for fruitful support.

Funding
Not applicable.

Availability of data and materials
The datasets (Iris, wine and heart disease) analysed during the current study are available in the [https://archive.ics.uci.edu/ml/index.php]. On the heart disease dataset, 13 of the best features is selected from the Dataset. [https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset]

Declarations

Competing interests
The authors declare no competing interests.

Author contributions
T.T. designed the research. S.N. and T.T. performed the calculations. T.T. and S.N. discussed the results. T.T. wrote the manuscript. T.T. and S.N. reviewed the manuscript. Both authors read and approved the final manuscript.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 15 July 2022 Accepted: 29 November 2022 Published online: 15 December 2022

References

1. Bartkiewicz K, Gneiting C, Černoch A et al. Experimental kernel-based quantum machine learning in finite feature space. Sci Rep. 2020;10:12356. https://doi.org/10.1038/s41598-020-68911-5.
2. Zaspel P, Huang B, Harbrecht H, Lilienfeld OA. Boosting quantum machine learning models with a multilevel combination technique: pople diagrams revisited. J Chem Theory Comput. 2019;15:1546–59. https://doi.org/10.1021/acs.jctc.8b00832.
3. Liu Y, Arunachalam S, Temme K. A rigorous and robust quantum speed-up in supervised machine learning. Nat Phys. 2021;17:1013–7. https://doi.org/10.1038/s41567-021-01287-z.
4. Johri S, Debnath S, Mocherla A et al. Nearest centroid classification on a trapped ion quantum computer. npj Quantum Inf. 2021;7:122. https://doi.org/10.1038/s41534-021-00456-5.
5. Wu SL et al. Application of quantum machine learning using the quantum variational classifier method to high energy physics analysis at the LHC on IBM quantum computer simulator and hardware with 10 qubits. J Phys G, Nucl Part Phys. 2021;48:125003. https://doi.org/10.1088/1361-6471/ac1391.
6. Sheykhmousa M, Mahdianpari M, Ghanbari H, Mohammadimanesh F, Ghamisi P, Homayouni S. Support vector machine versus random forest for remote sensing image classification: a meta-analysis and systematic review. IEEE J Sel Top Appl Earth Obs Remote Sens. 2020;13:6308–25. https://doi.org/10.1109/JSTARS.2020.3026724.
7. Panwar H, Gupta PK, Siddiqui MK, Morales-Menendez R, Singh V. Application of deep learning for fast detection of Covid-19 in X-rays using nCOVnet. Chaos Solitons Fractals. 2020;138:109944. https://doi.org/10.1016/j.chaos.2020.
8. Moen E, Bannon D, Kudo T et al. Deep learning for cellular image analysis. Nat Methods. 2019;16:1233–46. https://doi.org/10.1038/s41592-019-0403-1.
9. Khan S, Islam H, Jan Z, Din IU, Rodrigues JJPC. A novel deep learning based framework for the detection and classification of breast cancer using transfer learning. Pattern Recognit Lett. 2019;125:1–6. https://doi.org/10.1016/j.patrec.2019.03.022.
10. Liefeng B, Ren X, Fox D. Hierarchical matching pursuit for image classification: architecture and fast algorithms. Advances in neural information processing systems. 2011; 24. https://proceedings.neurips.cc/paper/2011.
11. Wuest T, Weimer D, Irgens C, Thoben KD. Machine learning in manufacturing: advantages, challenges, and applications. Prod Manuf Res. 2016;4:23–45. https://doi.org/10.1080/21693277.2016.1192517.
12. Wu M, Song Z, Moon YB. Detecting cyber-physical attacks in cyber manufacturing systems with machine learning methods. J Intell Manuf. 2019;30:1111–23. https://doi.org/10.1007/s10845-017-1315-5.
13. Kubat M, Holte RC, Matwin S. Machine learning for the detection of oil spills in satellite radar images. Mach Learn. 1998;30:195–215. https://doi.org/10.1023/A:1007452223027.
14. Liu P, Choo KKR, Wang L et al. SVM or deep learning? A comparative study on remote sensing image classification. Soft Comput. 2017;21:7053–65. https://doi.org/10.1007/s00500-016-2247-2.
15. Chapelle O, Haffner P, Vapnik VN. Support vector machines for histogram-based image classification. IEEE Trans Neural Netw. 1999;10:1055–64. https://doi.org/10.1109/72.788646.
16. Peña JM, Gutiérrez PA, Hervás-Martínez C, Six J, Plant RE, López-Granados F. Object-based image classification of summer crops with machine learning methods. Remote Sens. 2014;6:5019–41. https://doi.org/10.3390/rs6065019.
17. Shankar K, Lakshmanaprabu SK, Gupta D et al. Optimal feature-based multi-kernel SVM approach for thyroid disease classification. J Supercomput. 2020;76:1128–43. https://doi.org/10.1007/s11227-018-2469-4.
18. Bourouis S, Zaguia A, Bouguila N, Alroobaea N. Deriving probabilistic SVM kernels from flexible statistical mixture models and its application to retinal images classification. IEEE Access. 2019;7:1107–17. https://doi.org/10.1109/ACCESS.2018.2886315.
19. Altan A, Karasu S. The effect of kernel values in support vector machine to forecasting performance of financial time series. J Cogn Syst. 2019;4:17–21. https://dergipark.org.tr/en/pub/jcs/issue/44276/570863.

20. Havlíček V, Córcoles AD, Temme K et al. Supervised learning with quantum-enhanced feature spaces. Nature. 2019;567:209–12. https://doi.org/10.1038/s41586-019-0980-2.
21. Huang H-Y, Broughton M, Mohseni M et al. Power of data in quantum machine learning. Nat Commun. 2021;12:2631. https://doi.org/10.1038/s41467-021-22539-9.
22. Fawcett T. An introduction to ROC analysis. Pattern Recognit Lett. 2006;27:861–74. https://doi.org/10.1016/j.patrec.2005.10.010.
23. Davis J, Goadrich M. The relationship between precision-recall and ROC curves. In: Proc. 23rd inter. Conf. on machine learning (ICML'06). 2006. p. 233–40. https://doi.org/10.1145/1143844.1143874.
24. Tharwat A. Classification assessment methods. Appl Comput Inform. 2018;17:168–92. https://doi.org/10.1016/j.aci.2018.08.003.
25. Rahman MM, Antani SK, Thoma GR. A learning-based similarity fusion and filtering approach for biomedical image retrieval using SVM classification and relevance feedback. IEEE Trans Inf Technol Biomed. 2011;15:640–6. https://doi.org/10.1109/TITB.2011.2151258.
26. Mitarai K, Negoro N, Kitagawa M, Fujii K. Quantum circuit learning. Phys Rev A. 2018;98:032309. https://doi.org/10.1103/PhysRevA.98.032309.
27. Farhi E, Neven H. Classification with quantum neural networks on near term processors. 2018. arXiv:1802.06002.
28. Preskill J. Quantum computing in the NISQ era and beyond. Quantum. 2018;2:79–97. https://doi.org/10.22331/q-2018-08-06-79.