Received: 3 April 2024 | Revised: 2 July 2024 | Accepted: 13 August 2024

DOI: 10.4218/etrij.2024-0144

SPECIAL ISSUE

ETRI Journal WILEY

Metaheuristic optimization scheme for quantum kernel classifiers using entanglement-directed graphs

Yozef Tjandra | **Hendrik Santoso Sugiarto**

Calvin Institute of Technology, Jakarta, Indonesia

Correspondence
Hendrik Santoso Sugiarto, Calvin Institute of Technology, Jakarta, Indonesia.
Email: hendrik.sugiarto@calvin.ac.id

Funding information
This work was funded by PT Lancs Arche Consumma (MOU.003/CIT/XI/2022) and PT Astra International TBK - TSO (MOU.002/CIT/XI/2022).

Abstract

Entanglement is crucial for achieving quantum advantages. However, in the context of quantum machine learning, existing optimization strategies for generating quantum classifier circuits often result in unentangled circuits, indicating an underutilization of the entanglement effect needed to learn complex patterns. In this study, we proposed a novel metaheuristic approach—genetic algorithm—for designing a quantum kernel classifier that incorporates expressive entanglement. This classifier utilizes a loopless entanglement-directed graph, where each directed edge represents the entanglement between the target and control qubits. The proposed method consistently outperforms classical and quantum baselines across various artificial and actual datasets, achieving improvements up to 32.4% and 17.5%, respectively, compared with the best model among all other baselines. Moreover, this method successfully reconstructs the hidden entanglement structures underlying artificial datasets. The results also demonstrate that the optimized circuits exhibit diverse entanglement variations across different datasets, indicating the versatility of the proposed approach.

KEYWORDS
entanglement-directed graphs, genetic algorithm, optimization scheme, quantum kernels, quantum machine learning

1 | INTRODUCTION

Quantum computing promises a novel computational capacity that surpasses traditional classical computation. This capacity primarily arises from obscure quantum features such as superposition and entanglement. However, the current availability of quantum computers is limited to noisy intermediate-scale quantum (NISQ) frameworks [1], which restrict the potential of quantum computers to compete with more mature classical computers. Despite these limitations, a recent demonstration of quantum supremacy showed that even imperfect NISQ devices can significantly outperform classical computers in specific cases [2]. This progress has encouraged researchers to actively explore other practical areas of quantum advantages within NISQ frameworks [3, 4].

Quantum machine learning (QML) is a domain where quantum advantages can be utilized within NISQ devices [5, 6]. Numerous investigations have been conducted to extend classical machine-learning models to quantum computation frameworks. Various supervised classification models now have quantum equivalents, such as quantum k-nearest neighbors and quantum convolutional networks [7, 8].

The QML framework was proposed to enable more expressive pattern recognition by harnessing quantum computational advantages, which are difficult to replicate classically [9]. Within this framework, a variational method has been proposed, where a set of parameters is optimized within a quantum circuit to facilitate classification [10, 11]. Another alternative approach adopts a successful classical technique called the kernel support vector machine (SVM) [12, 13]. In the quantum version of this kernel method, each data point transforms the Hilbert space to enhance the expressive power of the space, aiding the SVM in constructing linear decision boundaries [14, 15]. Theoretical studies have suggested that both quantum and classical approaches are fundamentally equivalent when they reach their optimal form [16, 17]. Consequently, this study focuses on designing an optimal quantum kernel for SVM classification.

In this framework, data are first converted into a quantum state for processing by a quantum computer. A common conversion strategy involves designing a specific quantum circuit known as a quantum feature map. However, the rationale behind selecting various proposed maps remains unclear, suggesting the existence of potentially excellent but unexplored alternatives. Various studies have employed metaheuristic approaches to design optimal circuits for both variational and kernel methods [18, 19]. Interestingly, within this framework, most optimized circuits are not entangled, as the optimization schemes either provide minimal flexibility for subtle entanglement scheme design or employ ineffective gate choices.

Because entanglement is a key component of quantum advantages, these results suggest unexplored opportunities for utilizing entangled circuits in designing quantum feature maps. Addressing this gap, our research aims to design a new optimization scheme for discovering suitable entangled circuits. These circuits are configured as directed graphs, where the entanglement connections between qubit pairs can be either included or excluded. Therefore, our approach explores a more exhaustive entanglement graph combinatorial search space using a tailored metaheuristic optimization scheme.

The main contributions of this study are as follows: (i) the design of a combinatorial optimization scheme for identifying the most optimal Pauli feature map with entanglement digraphs to enhance classification performance, and (ii) an analysis of performance improvements compared with classical and quantum baselines.

2 | RELATED WORKS

This study explores the intersection of optimization methods and quantum circuit designs, focusing on multiple criteria within a machine-learning context. Specifically, we employ a metaheuristic optimization framework known as a genetic algorithm (GA), which mimics natural selection processes to identify the most suitable solutions across successive generations [20]. GAs are extensively applicable in numerous classical optimization domains, encompassing theoretical challenges, such as the metric dimension problem in graph theory and practical applications such as neural network architecture optimization in machine learning [21, 22].

GAs can be particularly beneficial for optimizing quantum circuits for diverse purposes. The design of optimal quantum circuits remains an open problem for various applications [23, 24]. In the QML context, GAs have been applied to automatically generate quantum feature maps to enhance SVMs [25]. The same authors later expanded their method by incorporating classical dimensionality reduction strategies to accommodate more intricate datasets [18]. Moreover, comparative analysis between SVMs enhanced by evolutionary-based kernels and the variational Ansatz method has been performed [26]. More recently, GAs have been employed to identify optimal Pauli feature maps for specific datasets [19].

Several alternative methods to GA optimization have also been proposed for automatically generating quantum circuits to enhance various machine-learning tasks. For example, a Parzen estimator has been proposed to identify optimal antenna circuits [27]. A Bayesian approach has recently been introduced to adaptively construct feature maps for SVM tasks [28]. Beyond the QML domain, GA are also applied to determine optimal circuit configurations within parametric quantum circuit design frameworks [29].

In contrast to the aforementioned approaches, this study focuses on incorporating the Pauli feature map family with more flexible entanglement graph arrangements and designing a metaheuristic scheme to generate such feature maps for specific datasets. This approach aims to fully utilize entanglement as the main feature of quantum advantage. Graph information is accommodated in various quantum-inspired models [30–32].

3 | METHODS

3.1 | Quantum kernel classifier

In a supervised machine-learning context, patterns and relationships between input data and their corresponding labels are learned using a trainable model. When dealing with complex data that cannot be separated linearly, a nonlinear classifier is required. The kernel SVM is an extensively used nonlinear classifier [12]. The kernel

SVM transforms the original dataset into a feature space with higher dimensions using a particular function, facilitating a linear separation between different classes.

The kernel SVM binary prediction for test data ($\vec{x}'$) is

$$\hat{y}(\vec{x}') = \text{sgn}\left(\sum_{m=1}^{M} c_m y_m K(\vec{x}_m, \vec{x}') + b\right), \quad (1)$$

where c_m and b denote the trainable parameters of the classical SVM, y_m represents the instance label of $\vec{x}_m$, and $K(\vec{x}_m, \vec{x}')$ denotes the kernel function between the training instance $\vec{x}_m$ and the test instance $\vec{x}'$. The RBF kernel $K(\vec{x}, \vec{z}) = \exp(-\gamma\|x-z\|^2)$ is a well-known example of a classical kernel owing to its good performance.

In the quantum setting, the kernel function is calculated using a specific quantum circuit that maps low-dimensional classical data into high-dimensional quantum states, enabling quantum interactions among data points. These quantum feature maps can be customized to capture specific data patterns, thereby improving model performance. The fundamental concept underlying quantum kernels is the efficient computation of the inner product of pairs of quantum data. These inner products are used to derive the feature kernel $K(\vec{x}, \vec{z}) = \left|\langle\Psi(\vec{x})|\Psi(\vec{z})\rangle\right|^2$. Quantum advantages can be harnessed by leveraging these quantum inner products [33].

3.1.1 | Quantum feature map

Hereafter, let n be the number of data features used in the classification. To obtain quantum data $|\Psi(\vec{x})\rangle$, a quantum feature map is used. Within a quantum feature map, the input data undergo transformation via specific unitary operations, resulting in a new quantum state vector that encompasses higher-order correlations among the original data points ($|\Psi(\vec{x})\rangle = U(\vec{x})|0\rangle^{\otimes n}$). Quantum feature maps can effectively generate intricate mappings that are difficult to compute using classical methods.

Quantum feature maps have demonstrated effectiveness across various machine-learning tasks, and many studies have focused on novel types of quantum feature maps and their practical applications in machine-learning problems [25, 33]. The ZZ feature map is a notable quantum feature map that is expensive to simulate classically that has yielded promising outcomes in several contexts [34, 35]. The quantum gate arrangement is also associated with the special case of a second-order Pauli feature map. Specifically, for the 2-qubit cases of Pauli feature maps based on the sequence $[Z, ZZ]$, the Pauli unitary expansion can be written as

$$U_{\phi(\vec{x})} = \exp(ix_0Z_0 + ix_1Z_1 + i(\pi - x_0)(\pi - x_1)Z_0Z_1). \quad (2)$$

The first two terms represent the R_Z rotation gate for each qubit: $\exp(ix_0Z_0) = R_Z(2x_0)$ and $\exp(ix_1Z_1) = R_Z(2x_1)$. The final tensor product term $\exp(i(\pi - x_0)(\pi - x_1)Z_0Z_1)$ can be rearranged into an entangled quantum circuit $CX \cdot (I \otimes R_Z(2(\pi - x_0)(\pi - x_1))) \cdot CX$.

3.1.2 | Augmented Pauli feature map with an entanglement-directed graph

The Pauli feature map can be further relaxed to allow greater flexibility in arranging two-qubit gates for any qubit pair. Specifically, we modify herein the original Pauli feature map into two types of unitary operators: rotation operators $R_{\Theta(\vec{x})}$ and entangled graph operators $E_{\Phi(\vec{x}),S}$, as follows:

$$U(\vec{x})|0\rangle^{\otimes n} = E_{\Phi(\vec{x}),S} H^{\otimes n} R_{\Theta(\vec{x})} H^{\otimes n} |0\rangle^{\otimes n}. \quad (3)$$

The rotation operators are defined as follows,

$$R_{\Theta(\vec{x})} = \exp\left(i \sum_{0 \le j < n} \theta_j(\vec{x}) P_j\right), \quad (4)$$

where $\theta_j(\vec{x}) = x_j$ and P_j is a rotation matrix chosen from the set $\{R_X, R_Y, R_Z, I\}$ that acts on the jth qubit. Thus, there are four choices for each qubit to select its rotation gate, resulting in 4^n possible configurations.

Next, the entangled Pauli expansion matrix $E_{\Phi(\vec{x})}$ is

$$E_{\Phi(\vec{x}),S} = \exp\left(i \sum_{(j,k) \in S} \phi_{jk}(\vec{x}) P_j P_k\right), \quad (5)$$

where $\phi_{jk}(\vec{x}) = (\pi - x_j)(\pi - x_k)$ serves as the quantum data interaction for features j and k, and set S represents a certain 2-qubit entanglement arrangement. Specifically, S consists of pairs of two distinct qubit indices (j, k), representing a two-qubit gate between the control qubit j and target qubit k involving Pauli matrices P_j and P_k. Note that our scheme follows the general form of the Pauli feature map by restricting the entanglement set S to include only 2-subsets of indices while generalizing the GA encoding scheme in PauliGA [19].

We note that the entanglement configuration for this operator can be summarized in the form of a loopless directed graph $G = (Q, S)$, where Q represents the set of

qubits as the vertex set, and S denotes the set of entanglement relationships as the directed edge set. Hence, there are $2^{n(n-1)}$ entanglement schemes. This directed graph structure was optimized to obtain the best-performing quantum feature map. To observe the subtle differences in the quantum circuits between entanglement schemes modeled by oppositely directed edges (i, j) and (j, i), as well as the variations in the order of Pauli matrices P_i and P_j, refer to Figure 1.

Furthermore, we provide an example of a quantum feature map for five features with a rotation operator given by

$$R = \exp(i(x_0X_0 + x_1Z_1 + x_2Y_2 + x_3I_3 + x_4Z_4))$$

and an entangled graph operator

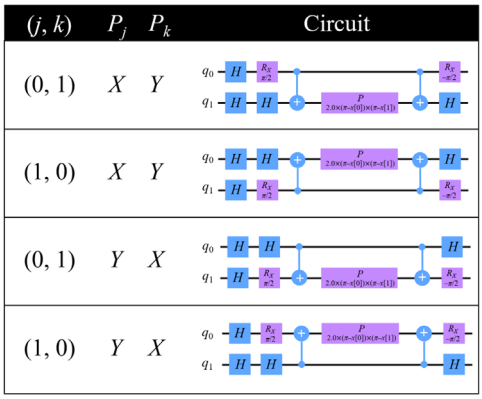

FIGURE 1 Listings of the (0,1) and (1,0) entanglements with $P_j, P_k \in \{X, Y\}$.

$$E = \exp\left(i \sum_{(j,k) \in S} (pi - x_j)(\pi - x_k) X_j Y_k\right)$$

with $S = \{(1, 0), (2, 0), (1, 3), (3, 2), (4, 0), (1, 4), (2, 4)\}$. The directed graph representation and the quantum circuit are shown in Figure 4.

3.2 | Genetic algorithm

The main method used in this study was metaheuristic; specifically, we adopted a GA to optimize the Pauli quantum feature map with an entanglement configuration to enhance the performance of the SVM. This method is known to overcome the challenge of local minima by combining random exploitation and exploration within the search space [20]. The proposed GA design, which utilizes the $(\mu + \lambda)$ strategy, is outlined in Figure 2. Additionally, the pseudocode for the overview algorithm is available in Algorithm 1 in supporting information Appendix B. Initially, the algorithm generates a population of random individuals, each encoded as a binary string representing an instance of the Pauli quantum feature map, which serves as a candidate solution. The algorithm then computes the fitness values for each individual in the population through multiple repetitions and, with certain probabilities, applies genetic operators (distinct from quantum operators) to select the fittest individual, representing the best solution to the optimization problem. Note that the likelihood of an individual's survival increases as a function of its fitness value.

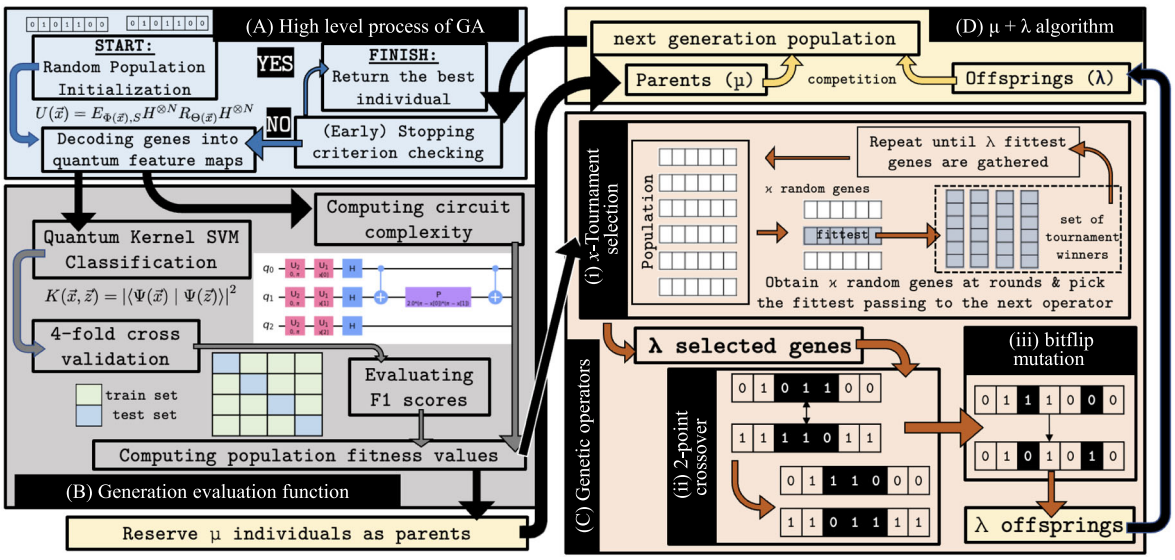

FIGURE 2 Overall description of the genetic algorithm (GA) used for generating high-performing quantum feature maps.

Genetic operators consist of a selection operator used for exploitation and crossover and mutation operators used for exploration. These are employed to produce λ new offsprings in each iteration.

This iterative process is repeated for a specific number of generations, which is a hyperparameter of the GA. Typically, the fittest individual from the previous generation is returned as the final solution. An additional evaluation of the final solution is conducted using a separate test case to determine its overall effectiveness of the GA solution.

3.2.1 | Genetic encoding

The quantum feature map design is based on three aspects: the entanglement digraph, initial rotation gates, and Pauli strings. Each of these components is encoded using (n^2-n), $2n$, and 4 binary bits, respectively. In other words, each gene in the population is a binary string with a total length of n^2+n+4. The entanglement digraph, which does not allow loops, is represented by directed edges denoted as (j, k) for some indices $j \neq k$. This results in n^2-n possible directed edges, each controlled by a single bit of the genetic code. For each qubit, an initial gate is selected from four possible choices (R_X, R_Y, R_Z), or an identity gate I, encoded using two binary bits. This allocation of $2n$ binary bits in the individual genes controls all initial rotation gates. The final 4 bits of the genetic code are employed to select the Pauli entanglement scheme, which is represented by a Pauli string of length two. Each digit in this string is chosen from the set $\{X, Y, Z, I\}$, with the encoding utilizing four binary bits. Figure 3 provides a detailed explanation of the technical interpretations of each block of bits, and Figure 4 provides an example illustrating of how the encoding works.

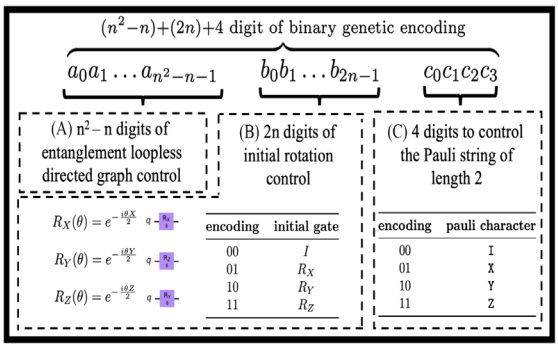

FIGURE 3 Genetic encoding scheme of the GA for the Pauli feature map family with an entanglement-directed graph.

3.2.2 | Fitness values

The versatility of GA encoding enables the representation of intricately entangled circuits, showcasing its potential to yield commendable classification performance. Nevertheless, as circuit complexity increases, managing error corrections for near-term quantum computers may become impractical. In this study, we constructed a fitness function designed to balance optimal classification performance with the avoidance of excessively complex circuits. An overview of this technique is shown in Figure 2B.

Specifically, we used the F1 score as the classification evaluation metric because it can tackle imbalanced datasets. In addition, in this study, we define circuit complexity as the number of gates involved in the circuit. Hence, to maximize classification performance and minimize circuit complexity, we define the fitness function as follows:

$$\text{Fitness} = \text{Complexity} + \frac{w}{\text{F1-Score}}, \tag{6}$$

where w is a positive constant selected experimentally to balance the two objectives.

3.2.3 | Genetic operators

An overview of the functioning of genetic operators is shown in Figure 2C. In the context of κtournament selection, κ random individuals from the population are

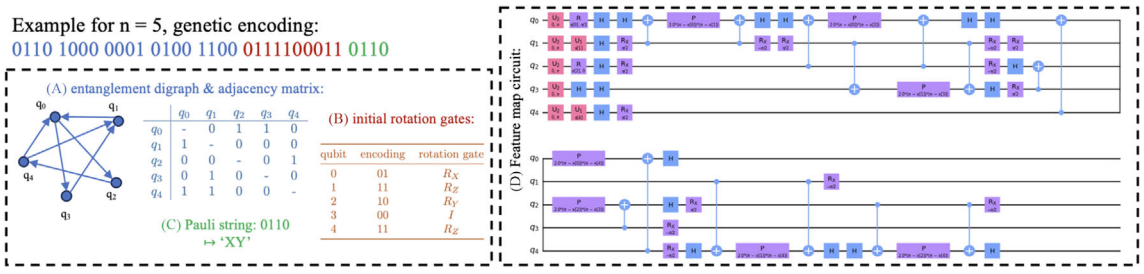

FIGURE 4 Example of decoding a genetic code into a quantum feature map.

chosen to compete in the tournament. The individuals with the highest fitness values won the tournament and advanced to subsequent genetic operators. In this study, a value of $\kappa = 5$ was used. The selected parents were then randomly paired, and a crossover occurred with a certain probability. This involved choosing two random indices to define an interval for gene exchange between parent pairs (see Figure 2C, ii). Each individual also had a chance to mutate, and some genes flipped with a fixed probability (see Figure 2C, iii). These probabilities were treated as hyperparameters of the GA model and were fine-tuned experimentally for optimal performance.

4 | DATA

4.1 | Datasets

To assess the effectiveness of the proposed method, it was tested on several artificial and actual datasets. We designed artificial datasets such that data points with different labels can be geometrically separated by a feature map with a specific entanglement graph, thus modifying the scheme introduced in the original quantum kernel publication [14]. Technically, we assigned binary classes $\hat{y}$ to the data features $\vec{x} \in [0,\ 2\pi]^n$, that is,

$$\hat{y}(\vec{x}) = \begin{cases} 1, & \text{if} \langle \Psi(\vec{x}) | V^\dagger M V | \Psi(\vec{x}) \rangle > \Delta, \\ 0, & \text{if} \langle \Psi(\vec{x}) | V^\dagger M V | \Psi(\vec{x}) \rangle < -\Delta, \end{cases} \tag{7}$$

where the expectation values require a random unitary vector $V \in SU(2^n)$ and a majority operator M with a Boolean function ($f : \{0,1\}^n \mapsto \{0,1\}$) that maps to class 1 if more than half of the bit-string digits are "1," and to class 0 otherwise. The observable $V^\dagger M V$ is then operated in the predefined feature maps $|\Psi(\vec{x})\rangle = U(\vec{x})|0\rangle^{\otimes n}$ with $U(\vec{x})$ as stated in (3), using specific entanglement schemes. Furthermore, a particular real-valued gap, $\Delta = 0.3$ between the expected values was chosen for data separation.

We generated six artificial datasets with various features and entanglement structures. The parameters of these datasets are listed in Table 1. The Z and ZZ operators are intentionally chosen as the initial rotation and entangled graph operators, respectively, to facilitate a meaningful comparison of our method with ZZ feature map baselines.

Moreover, we tested the performance of the proposed method on several small real-world datasets with varying numbers of features (Table 2). Access to these datasets was facilitated using the OpenML Python Package [36]. Details for each dataset used in the study are presented in Tables 1and 2.

4.2 | Data preprocessing and validation

To prepare for training, each numerical feature was normalized, whereas categorical data features were mapped into discretized numerical points within the normalized interval. The dataset was then partitioned into an 80% training set and a 20% testing set. To avoid overfitting, we further divided the training set into four groups using a 4-fold cross-validation approach. Within each group, the model underwent training and testing, yielding two key

TABLE 1 Feature map parameters used for generating artificial datasets.

Dataset	No. of features	Entanglement directed edges	Initial rotation	Pauli string
3-path	3	{(0, 1), (1, 2)}	*ZZZ*	["*ZZ*"]
3-clique	3	{(0, 1), (0, 2), (1, 2)}	*ZZZ*	["*ZZ*"]
3-biclique	3	{(0, 1), (0, 2), (1, 0), (1, 2), (2, 0), (2, 2)}	ZZZ	["*ZZ*"]
4-path	4	{(0, 1), (1, 2), (2, 3)}	*ZZZZ*	["*ZZ*"]
4-clique	4	{(0, 1), (0, 2), (0, 3), (1, 2), (1, 3), (2, 3)}	*ZZZZ*	["*ZZ*"]
4-biclique	4	{(0, 1), (0, 2), (0, 3), (1, 0), (1, 2), (1, 3), (2, 0), (2, 1), (2, 3), (3, 0), (3, 1), (3, 2)}	*ZZZZ*	["*ZZ*"]

TABLE 2 Specifications of actual datasets.

Dataset	Iris	Blood	Irish	Veteran	Liver	Telescope
Source	[37]	[38]	[39]	[40]	[41]	[42]
No. of features	4	4	5	7	10	10
No. of instances	150	748	500	137	583	500
No. of labels	3	2	2	2	2	2

classification evaluation metrics: accuracy and F1 score (macro-averaged). The average scores from the 4-fold groups were used as fitness values for the GA. This 4-fold scheme ensures that the feature map obtained during training does not overly cater to a specific subset of the training set while performing poorly on others. After the GA determines the best feature map, it uses the entire 80% of the training set for the final training iteration. The overall evaluation of the model was then determined using the remaining 20% of the test set, which remained unseen by the model during the evolutionary training process.

5 | RESULTS AND DISCUSSION

5.1 | Experimental settings

In this section, we assess the effectiveness of our method in generating quantum feature maps for both artificial and actual datasets. The GA hyperparameters used in the study are listed in Table 3. These values were further modified and refined through experimental adjustments based on other GA realizations [19, 25]. The construction of the quantum circuit described in this study was implemented on classical computers using the IBM-Qiskit framework with an Aer Backend [43].

We selected four classical, eight quantum, and one quantum-GA classifier for comparison with our method. For the classical baselines, we employed an SVM model with linear, polynomial, sigmoid, and RBF kernels. For the quantum baselines, we considered two basic quantum classifier paradigms: a quantum kernel SVM (*QSVC*) and a variational quantum classifier (*VQC*). The (*QSVC*) employs four basic circuits derived from the Pauli feature map family (*X*, *Y*, *Z*, and *ZZ* feature maps) and two recently proposed entangled kernels in linear (*ent-QSVC(lin)*) and full (*ent-QSVC(full)*) fashion [44]. The (*VQC*) employs two types of quantum circuits: hardware-efficient Ansatz (*HEA*) and modified hardware-efficient Ansatz (*MHEA*) [45]. Examples of these circuits are provided in the supporting information Appendix C. Finally, the GA-assisted quantum kernel baseline was obtained using the Pauli-GA scheme [19].

TABLE 3 GA hyperparameters used in the optimization process.

Hyperparameter	Value
Population size (μ)	100
Offspring size (λ)	60
Number of generations	200
Crossover probability	0.3
Individual mutation probability	0.7
Bitflip mutation probability	0.25

The results for the artificial and actual datasets are listed in Tables 4 and 5. All the performance metrics were derived from the test dataset, which comprised instances that were not included in the training process. The highest metric scores for each dataset are highlighted in bold, whereas the best scores for each baseline group are shown in italics. We also quantified the improvements obtained by the proposed method over the best among all the baselines. A summary of the best feature maps produced by the proposed method is presented in Table 6.

5.2 | Model performance in artificial datasets

Based on comparisons across all six proposed artificial datasets, our method consistently and significantly outperformed the other classical and quantum baselines yielding a 32.4% improvement, as shown in Table 4. Some of the best classical baselines, such as the polynomial and RBF kernels, can distinguish 3-feature entanglement combinations but struggle with more complex patterns like 4-feature entangled combinations. This is reasonable as the edge configuration of a 3-vertex graph is much simpler and more limited than that of larger graphs. For the quantum baselines, the entangled-QSVC kernels (both linear and full) tend to outperform the other baselines owing to their entanglement structures. However, even their best performances are still inferior to our method's performance. This is likely because our approach offers greater flexibility in adjusting to any hidden entanglement structures within the dataset, in contrast to the fixed entangled quantum baselines.

By further investigating the entanglement patterns generated by the best individuals using our method (Table 6), we found that they closely resembled the original operators and entanglement digraphs underlying the artificial datasets. For the 3-feature datasets, the best-generated individuals were exactly matched. There were slight differences, particularly in terms of the initial rotation choices in the artificial 4-feature datasets. By observing 4-path and 4-biclique datasets, our method perfectly captures the original entanglement digraphs used to generate the data. However, the digraph produced by the proposed GA method for the 4-clique dataset only slightly differs in terms of the edge directions. This signifies the ability of the proposed GA method to discover effectively specific entangling patterns that suit the classification tasks on particular datasets.

TABLE 4 Accuracies and F1-scores of classification on artificial datasets.

		ACC						F1					
Group	**Model**	**3-path**	**3-clique**	**3-biclique**	**4-path**	**4-clique**	**4-biclique**	**3-path**	**3-clique**	**3-biclique**	**4-path**	**4-clique**	**4-biclique**
Classical	SVC-linear	0.750	0.550	0.300	0.450	*0.550*	0.350	0.744	0.520	0.293	0.449	0.540	0.348
	SVC-poly	0.750	*0.800*	*0.850*	*0.650*	0.450	0.500	0.744	*0.798*	*0.847*	*0.642*	0.437	0.500
	SVC-rbf	*0.900*	*0.800*	0.800	0.550	*0.550*	*0.600*	*0.896*	0.792	0.792	0.549	*0.549*	*0.600*
	SVC-sigmoid	0.650	0.400	0.450	0.400	0.450	0.350	0.561	0.400	0.373	0.286	0.413	0.307
Quantum	QSVC-*X*	0.450	0.450	0.450	0.450	0.450	*0.450*	0.310	0.310	0.310	0.310	0.310	0.310
	QSVC-*Y*	0.350	0.600	0.250	0.450	0.500	0.400	0.348	0.596	0.233	0.449	0.500	0.394
	QSVC-*Z*	0.350	0.600	0.250	0.450	0.500	0.400	0.348	0.596	0.233	0.449	0.500	0.394
	QSVC-*ZZ*	0.600	0.650	0.500	0.550	0.600	0.450	0.600	0.642	0.495	0.549	0.600	0.449
	ent-QSVC(lin)	*0.750*	*0.750*	0.550	*0.900*	*0.650*	0.500	*0.749*	*0.749*	0.540	*0.899*	*0.627*	0.500
	ent-QSVC(full)	*0.750*	0.700	*0.650*	0.600	0.600	*0.600*	0.733	0.697	*0.627*	0.600	0.596	*0.596*
	VQC-HEA	0.700	*0.750*	0.350	0.600	0.500	*0.600*	0.700	0.715	0.348	0.583	0.479	*0.596*
	VQC-MHEA	0.650	0.500	0.500	0.450	0.600	0.450	0.649	0.495	0.495	0.437	0.600	0.437
qGA	Pauli-GA	0.750	0.650	0.850	0.700	0.700	0.650	0.744	0.642	0.840	0.688	0.670	0.642
	Proposed method	***1.000***	***1.000***	***1.000***	***0.950***	***0.950***	***0.950***	***1.000***	***1.000***	***1.000***	***0.949***	***0.949***	***0.950***
improv.	(Over the best baseline)	↑10%	↑20%	↑15%	↑5.3%	↑26%	↑31.6%	↑10.4%	↑20%	↑15%	↑5.3%	↑29%	↑32.4%

TABLE 5 Accuracies and F1-scores of classification on actual datasets.

		ACC						F1					
Group	**Model**	**Iris**	**Blood**	**Irish**	**Veteran**	**Liver**	**Telescope**	**Iris**	**Blood**	**Irish**	**Veteran**	**Liver**	**Telescope**
Classical	SVC-linear	*0.967*	0.733	*0.830*	0.714	*0.700*	0.730	0.958	0.467	*0.830*	*0.619*	0.412	0.675
	SVC-poly	0.933	*0.740*	0.755	0.714	*0.700*	*0.820*	0.919	0.449	0.755	0.513	0.412	*0.777*
	SVC-rbf	*0.967*	*0.740*	0.766	0.679	*0.700*	0.810	*0.958*	0.471	0.762	0.404	0.412	0.762
	SVC-sigmoid	0.467	0.690	0.574	*0.750*	0.620	0.700	0.440	*0.592*	0.571	0.537	*0.506*	0.412
Quantum	QSVC-*X*	0.933	*0.740*	0.766	0.679	0.700	0.700	0.919	0.449	0.762	0.404	0.412	0.412
	QSVC-*Y*	0.200	0.733	0.543	*0.750*	0.700	0.810	0.111	0.423	0.352	0.429	0.412	0.762
	QSVC-*Z*	*0.967*	*0.740*	0.766	0.679	0.700	0.810	*0.958*	0.449	0.762	0.404	0.412	0.762
	QSVC-*ZZ*	0.833	0.733	0.904	0.714	0.620	0.810	0.822	0.446	0.904	0.417	0.479	0.767
	ent-QSVC(lin)	*0.967*	0.733	*0.968*	0.678	*0.701*	*0.840*	*0.958*	*0.467*	*0.968*	*0.491*	0.482	*0.808*
	ent-QSVC(full)	*0.967*	0.733	*0.968*	0.678	*0.701*	0.815	*0.958*	0.423	*0.968*	0.404	0.482	0.774
	VQC-HEA	0.867	0.727	0.723	0.607	0.649	0.685	0.864	0.421	0.713	0.449	0.513	0.435
	VQC-MHEA	0.933	*0.740*	0.766	0.607	0.641	0.685	0.931	0.449	0.764	0.449	*0.521*	0.449
qGA	Pauli-GA	0.967	0.760	0.915	0.750	0.640	0.830	0.958	0.615	0.915	0.604	0.554	0.787
	proposed method	***1.000***	***0.800***	***1.000***	***0.821***	***0.720***	***0.850***	***1.000***	***0.628***	***1.000***	***0.750***	***0.592***	***0.816***
improv.	(Over the best baseline)	↑3.3%	↑5.0%	↑3.2%	↑8.6%	↑2.6%	↑1.2%	↑4.2%	↑2.1%	↑3.2%	↑17.5%	↑6.4%	↑1.0%

TABLE 6 Best individual circuit parameters among all tested datasets.

Group	Dataset	No. of features	Entanglement_scheme	Initial_rotation	Pauli_string
artificial	3-path	3	[(0, 1), (1, 2)]	ZZZ	["ZZ"]
	3-clique	3	[(0, 1), (2, 0), (1, 2)]	ZZZ	["ZZ"]
	3-biclique	3	[(0, 1), (1, 0), (2, 0), (0, 2), (1, 2), (2, 1)]	ZZZ	["ZZ"]
	4-path	4	[(0, 1), (1, 2), (2, 3)]	ZYYZ	["ZZ"]
	4-clique	4	[(0, 1), (2, 0), (3, 0), (1, 2), (3, 1), (2, 3)]	ZZZY	["ZZ"]
	4-biclique	4	[(0, 1), (1, 0), (2, 0), (3, 0), (0, 2), (1, 2), (2, 1), (3, 1), (0, 3), (1, 3), (2, 3), (3, 2)]	*YZZZ*	["*ZZ*"]
actual	iris	4	[(0, 1), (1, 0), (2, 1), (1, 3), (2, 3), (3, 2)]	*ZIYZ*	["*IX*"]
	blood	4	[(1, 0), (2, 0), (3, 0), (0, 2), (1, 2), (2, 1), (0, 3), (2, 3), (3, 2)]	*ZZZY*	["*YZ*"]
	irish	5	[(0, 3), (4, 2), (2, 4)]	*ZYYIX*	["*IY*"]
	veteran	7	[(0, 1), (2, 0), (3, 0), (0, 2), (1, 2), (1, 3), (3, 2), (4, 2), (5, 2), (0, 4), (1, 4), (3, 4), (0, 5), (2, 5), (3, 5), (4, 5), (5, 4), (3, 6), (6, 5)]	*ZZYYZZY*	["*ZI*"]
	liver	10	[(3, 0), (4, 0), (6, 0), (7, 0), (0, 2), (3, 1), (1, 3), (5, 2), (6, 2), (7, 2), (8, 2), (4, 3), (5, 3), (7, 3), (1, 5), (2, 5), (3, 5), (6, 4), (1, 6), (2, 6), (3, 6), (5, 6), (7, 5), (4, 7), (5, 7), (0, 9), (2, 9), (4, 9), (5, 9), (8, 9)]	*XXXIZXIXZX*	["*YI*"]
	telescope	10	[(0, 1), (1, 0), (2, 0), (6, 0), (4, 1), (5, 1), (6, 1), (2, 3), (3, 2), (4, 2), (5, 2), (8, 2), (0, 4), (3, 5), (5, 4), (6, 4), (0, 6), (1, 6), (2, 6), (4, 6), (0, 7), (1, 7), (0, 8), (5, 8), (7, 8), (9, 7)]	*YXYXZZIZIZ*	["*ZY*"]

5.3 | Model performance in actual datasets

Similarly, across the six actual datasets, our method outperformed other classical and quantum baselines, as shown in Table 5. Among the baselines, the unentangled kernels generally performed the poorest, while the best-performing models were either classical SVC or quantum classifiers involving specific entanglement schemes. However, the GA-crafted quantum kernels yielded significantly better results, with improvements of up to 17.5% compared with the best baseline performances. This indicates that enhancing the performance of a QSVC for a particular dataset requires modifying the subtle architecture of the quantum feature maps rather than relying on regular patterns provided by quantum kernel baselines.

Notably, among the six optimal circuits generated by our method from each actual dataset, only two—those from the"blood" and "telescope" datasets—included entangling gates. Although these are not the primary results, we briefly report that none of the best circuits obtained from other Pauli-GA baselines involved entanglements, further demonstrating the expressiveness of our method. Although some entangling circuits may perform as well as non-entangling alternatives, the GA scheme might penalize these circuits due to their complexity. In such cases, entangled circuits may not be necessary to achieve good performance.

Additionally, our method's ability to freely choose the entanglement combinations of the Pauli feature maps (the Pauli string choices) contributes significantly to identifying a classifier with improved performance. While all other quantum baselines rely on the *ZZ*-type entanglement owing to its versatility in various contexts, our method can discover alternative entanglement types. This is evident from the final GA-generated entangling quantum feature maps for the "blood" and "telescope" datasets, which do not follow the *ZZ*-type entanglement form. This suggests that the Pauli string is unsuitable for these specific datasets, but our method can identify a more effective configuration.

Whether using an entangling or not, the flexibility of our method in selecting entangling qubits plays a crucial role in enhancing classification results. Even without entangling the gates, our method outperformed the Pauli GA baseline alternative. However, while our method can discover entangled feature maps, the Pauli GA baseline cannot find such configurations. This highlights the proposed method's ability to detect a suitable quantum feature map with a more defined entanglement configuration, leading to improved classification performance for specific datasets.

To assess the reliability of our method on NISQ devices, we provide an upper bound on the number of qubits for which our GA-generated circuits remain feasible. Assuming state-of-the-art error rates of 10^{-4} [46], we

estimate that circuits produced by our method with fewer than 13 qubits can handle noise even in worst-case entanglement scenarios. Because all datasets in this study have no more than 10 features, our results are moderately reliable for current quantum devices. Additionally, our fitness function design optimizes for smaller circuits, further increasing the resilience of NISQ devices. More detailed estimations and further discussions on this topic are provided in the supporting information Appendix D.

6 | CONCLUSIONS AND FUTURE WORKS

This study applied a metaheuristic optimization scheme to produce quantum feature maps with flexible entanglement arrangements. Using various actual and artificial datasets, our proposed method consistently outperformed other classical, quantum, and quantum-GA baselines. Compared with the best-performing baselines, our method achieved improvements ranging from 5.3% to 32.4% for artificial datasets and 1.0% to 17.5% for actual datasets. This method successfully identifies entanglement patterns that better accommodate data than classical classifiers and other quantum approaches, such as ZZ feature maps, entangled-QSVC, QSVC, and maps generated by Pauli-GA baselines. For artificial datasets, the proposed method finds entanglement structures similar to the predefined entanglement graph patterns used in data construction. In actual datasets, the generated feature map does not converge toward specific entanglements and can even generate quantum circuits without entanglement when unnecessary. This indicates the method's capability to discover diverse entangled feature maps tailored to each dataset's unique characteristics. However, as the dimensions of the datasets increase, the performance of this method may degrade when applied to current NISQ processors. This could be a significant limitation when dealing with more complex datasets, such as those involving text and images. Several potential directions for future research and improvement exist. Further exploration is necessary to understand why certain entanglement structures enhance performance for specific datasets. Additional measures should be taken to identify the roles of specific entanglements in various cases. Refining the entanglement styles to encompass a broader array of options, such as incorporating higher-order quantum circuit interactions like N-local alternating entanglement layers, could further boost performance. Moreover, this optimization framework has the potential for broader applications, extending beyond the machine-learning domain to optimize quantum circuits in other fields.

CONFLICT OF INTEREST STATEMENT

The authors declare no potential conflicts of interest.

ORCID

Yozef Tjandra https://orcid.org/0000-0001-8756-9176
Hendrik Santoso Sugiarto https://orcid.org/0000-0002-1831-9808

REFERENCES

1. J. Preskill, *Quantum computing in the NISQ era and beyond*, Quantum **2** (2018), 79. https://doi.org/10.22331/q-2018-08-06-79
2. F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell, and B. Burkett, *Quantum supremacy using a programmable superconducting processor*, Nature **574** (2019), no. 7779, 505–510. https://doi.org/10.1038/s41586-019-1666-5
3. S. Bravyi, D. Gosset, R. König, and M. Tomamichel, *Quantum advantage with noisy shallow circuits*, Nature Phys. **16** (2020), no. 10, 1040–1045. https://doi.org/10.1038/s41567-020-0948-z
4. M. R. Perelshtein, A. I. Pakhomchik, A. A. Melnikov, M. Podobrii, A. Termanova, I. Kreidich, B. Nuriev, S. Iudin, C. W. Mansell, and V. M. Vinokur, *NISQ-compatible approximate quantum algorithm for unconstrained and constrained discrete optimization*, Quantum **7** (2023), 1186. https://doi.org/10.22331/q-2023-11-21-1186
5. Y. Wang and J. Liu, *Quantum machine learning: from NISQ to fault tolerance*, arXiv preprint, 2024. https://doi.org/10.48550/arXiv.2401.11351
6. Y. Zhang and Q. Ni, *Recent advances in quantum machine learning*, Quantum Eng. **2** (2020), no. 1, e34.
7. I. Cong, S. Choi, and M. D. Lukin, *Quantum convolutional neural networks*, Nat. Phys. **15** (2019), no. 12, 1273–1278.
8. Y. Dang, N. Jiang, H. Hu, Z. Ji, and W. Zhang, *Image classification based on quantum k-nearest-neighbor algorithm*, Quantum Inf. Process. **17** (2018), 1–18.
9. M. J. Bremner, A. Montanaro, and D. J. Shepherd, *Achieving quantum supremacy with sparse and noisy commuting quantum computations*, Quantum **1** (2017), 8.
10. K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, *Quantum circuit learning*, Phys. Rev. A **98** (2018), no. 3, 32309.
11. M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, *Circuit-centric quantum classifiers*, Phys. Rev. A **101** (2020), no. 3, 32308.
12. C. Cortes and V. N. Vapnik, *Support-vector networks*, Mach. Learn. **20** (2004), 273–297.
13. S. Theodoridis and K. Koutroumbas, *Pattern recognition*, Elsevier, 2006.
14. 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 **567** (2019), no. 7747, 209–212.
15. M. Schuld and N. Killoran, *Quantum machine learning in feature Hilbert spaces*, Phys. Rev. Lett. **122** (2019), no. 4, 40504.
16. S. Jerbi, L. J. Fiderer, H. Poulsen Nautrup, J. M. Kübler, H. J. Briegel, and V. Dunjko, *Quantum machine learning beyond*

kernel methods, Nature Commun. **14** (2023), no. 1, 517. https://doi.org/10.1038/s41467-023-36159-y
17. M. Schuld, *Supervised quantum machine learning models are kernel methods*, arXiv preprint, 2021. https://doi.org/10.48550/arXiv.2101.11020
18. S. Altares-López, J. J. García-Ripoll, and A. Ribeiro, *AutoQML: Automatic generation and training of robust quantum-inspired classifiers by using genetic algorithms on grayscale images*, arXiv preprint, 2022. https://doi.org/10.48550/arXiv.2208.13246
19. Y. Tjandra and H. Sugiarto, *An evolutionary algorithm design for Pauli-based quantum kernel classification*, (Joint Workshops 49th Int. Conf. Very Large Data Bases – Int. Workshop Quantum Data Sci. Manag., Cancouver, Canada), 2023.
20. L. Davis, Handbook of genetic algorithms, 1991.
21. H. Chiroma, S. Abdulkareem, A. Abubakar, and T. Herawan, *Neural networks optimization through genetic algorithm searches: a review*, Appl. Math. Inf. Sci **11** (2017), no. 6, 1543–1564.
22. J. Kratica, V. Kovačević-Vujčić, and M. Čangalović, *Computing the metric dimension of graphs by genetic algorithms*, Computat. Optim. Applicat. **44** (2009), no. 2, 343–361.
23. T. Hubregtsen, J. Pichlmeier, P. Stecher, and K. Bertels, *Evaluation of parameterized quantum circuits: on the relation between classification accuracy, expressibility, and entangling capability*, Quantum Mach. Intell. **3** (2021), no. 1, 9. https://doi.org/10.1007/s42484-021-00038-w
24. S. Sim, P. D. Johnson, and A. Aspuru-Guzik, *Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms*, Adv. Quantum Technol. **2** (2019), no. 12, 1900070. https://doi.org/10.1002/qute.201900070
25. S. Altares-López, A. Ribeiro, and J. J. García-Ripoll, *Automatic design of quantum feature maps*, Quantum Sci. Technol. **6** (2021), no. 4, 45015.
26. B.-S. Chen and J.-L. Chern, *Genetically auto-generated quantum feature maps*, arXiv preprint, 2022.
27. N. Nguyen and K.-C. Chen, *Quantum embedding search for quantum machine learning*, IEEE Access **10** (2022), 41444–41456.
28. E. Torabian and R. V. Krems, *Compositional optimization of quantum circuits for quantum kernels of support vector machines*, Phys. Rev. Res. **5** (2023), no. 1, 13211.
29. M. Incudini, F. Martini, and A. Di Pierro, *Structure learning of quantum embeddings*, arXiv preprint, 2022. https://doi.org/10.48550/arXiv.2209.11144
30. L. Bai, L. Cui, Y. Wang, M. Li, J. Li, S. Y. Philip, and E. R. Hancock, *HAQJSK: hierarchical-aligned quantum Jensen-Shannon kernels for graph classification*, IEEE Trans. Knowl. Data Eng. (2024).
31. L. Cui, M. Li, L. Bai, Y. Wang, J. Li, Y. Wang, Z. Li, Y. Chen, and E. R. Hancock, *QBER: quantum-based entropic representations for un-attributed graphs*, Pattern Recogn. **145** (2024), 109877. https://doi.org/10.1016/j.patcog.2023.109877
32. Q. Meng, J. Zhang, Z. Li, M. Li, and L. Cui, *Entangled quantum neural network*, in*Quantum computing: a shift from bits to qubits*, Springer, 2023, pp. 245–262.
33. Y. Suzuki, H. Yano, Q. Gao, S. Uno, T. Tanaka, M. Akiyama, and N. Yamamoto, *Analysis and synthesis of feature map for kernel-based quantum classifier*, Quantum Mach. Intell. **2** (2020), 1–9.
34. M. Grossi, N. Ibrahim, V. Radescu, R. Loredo, K. Voigt, C. Von Altrock, and A. Rudnik, *Mixed quantum–classical method for fraud detection with quantum feature selection*, IEEE Trans. Quantum Eng. **3** (2022), 1–12.
35. J. Mancilla and C. Pere, *A preprocessing perspective for quantum machine learning classification advantage in finance using nisq algorithms*, Entropy **24** (2022), no. 11, 1656.
36. M. Feurer, J. N. Van Rijn, A. Kadra, P. Gijsbers, N. Mallik, S. Ravi, A. Müller, J. Vanschoren, and F. Hutter, *Openml-python: an extensible python api for openml*, J. Mach. Learn. Res. **22** (2021), no. 1, 4573–4577.
37. R. A. Fisher, *The use of multiple measurements in taxonomic problems*, Ann. Eugen. **7** (1936), no. 2, 179–188.
38. I.-C. Yeh, K.-J. Yang, and T.-M. Ting, *Knowledge discovery on rfm model using Bernoulli sequence*, Expert Syst. Appl. **36** (2009), no. 3, 5866–5871.
39. V. Greaney and T. Kellaghan, *Equality of opportunity in irish schools: a longitudinal study of 500 students*, Educational Company, 1984.
40. J. D. Kalbfleisch and R. L. Prentice, *The statistical analysis of failure time data*, John Wiley & Sons, 2011.
41. B. V. Ramana, M. S. P. Babu, and N. B. Venkateswarlu, *A critical comparative study of liver patients from USA and India: an exploratory analysis*, Int. J. Comput. Sci. Issues **9** (2012), no. 3, 506–516.
42. R. K. Bock, A. Chilingarian, M. Gaug, F. Hakl, T. Hengstebeck, M. Jiřina, J. Klaschka, E. Kotrč, P. Savický, S. Towers, and A. Vaiciulis, *Methods for multidimensional event classification: a case study using images from a Cherenkov gamma-ray telescope*, Nucl. Instrum. Methods Phys. Res. Sect. A: Accelerators, Spectrometers, Detectors and Assoc. Equip. **516** (2004), no. 2-3, 511–528.
43. Qiskit contributors, *Qiskit: an open-source framework for quantum computing*, 2023.
44. D. Sharma, P. Singh, and A. Kumar, *The role of entanglement for enhancing the efficiency of quantum kernels towards classification*, Phys. A: Stat. Mechan. Applicat. **625** (2023), 128938. https://doi.org/10.1016/j.physa.2023.128938
45. D. A. Shoieb, A. Younes, S. M. Youssef, and K. M. Fathalla, *HQMC-CPC: a hybrid quantum multiclass cardiac pathologies classification integrating a modified hardware efficient ansatz*, IEEE Access **12** (2024), 18295–18314.
46. Z. Li, P. Liu, P. Zhao, Z. Mi, H. Xu, X. Liang, T. Su, W. Sun, G. Xue, J. N. Zhang, and W. Liu, *Error per single-qubit gate below 10-4 in a superconducting qubit*, npj Quantum Inform. **9** (2023), no. 1, 111. https://doi.org/10.1038/s41534-023-00781-x
47. *Suppressing quantum errors by scaling a surface code logical qubit*, Nature **614** (2023), no. 7949, 676–681.
48. S. Brandhofer, S. Devitt, T. Wellens, and I. Polian, Special session: noisy intermediate-scale quantum (NISQ) computers–how they work, how they fail, how to test them? 2021 IEEE 39th VLSI Test Symposium (VTS), San Diego, CA, USA, 2021, pp. 1–10. https://doi.org/10.1109/VTS50974.2021.9441047
49. G. González-García, R. Trivedi, and J. I. Cirac, *Error propagation in NISQ devices for solving classical optimization problems*, PRX Quantum **3** (2022), no. 4, 40326. https://doi.org/10.1103/PRXQuantum.3.040326

50. F. Leymann and J. Barzen, *The bitter truth about gate-based quantum algorithms in the NISQ era*, Quantum Sci. Technol. **5** (2020), no. 4, 44007. https://doi.org/10.1088/2058-9565/abae7d

AUTHOR BIOGRAPHIES

Yozef Tjandra received his Master's degree in Mathematics from Monash University, Australia, focusing on probabilistic combinatorics. Since 2019, he has been a Lecturer in the IT and Big Data Analytics Department of the Calvin Institute of Technology, Indonesia. He has been working on various mathematical and computation-related projects for educational and commercial purposes. His research interests include enumerative combinatorics, machine-learning applications, optimization, and quantum machine learning.

Hendrik Santoso Sugiarto obtained his Ph.D. degree in Physics of Complex Systems from Nanyang Technological University, Singapore. His research interests include phase transition (in nature and society), nonlinear dynamics, complex networks, probabilistic graphical modeling, deep learning, and quantum computing. He has published several research articles on these topics. He has worked as a research scientist at a smart national research center in Singapore and as a data scientist at a leading startup company in Indonesia. He has been dealing with several machine learning and data science projects on various topics, such as the labor market, default risk, and fraud networks. He is currently an academic faculty member at the Calvin Institute of Technology in Indonesia.

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Y. Tjandra and H. S. Sugiarto, *Metaheuristic optimization scheme for quantum kernel classifiers using entanglement-directed graphs*, ETRI Journal **46** (2024), 793–805, DOI 10.4218/etrij.2024-0144