IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 27 May 2022, accepted 4 July 2022, date of publication 14 July 2022, date of current version 25 July 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3190897*

RESEARCH ARTICLE

# Integrating Machine Learning Algorithms With Quantum Annealing Solvers for Online Fraud Detection

HAIBO WANG [1], WENDY WANG [2], YI LIU [3], AND BAHRAM ALIDAEE [4]

[1] Division of International Business and Technology Studies, Texas A&M International University, Laredo, TX 78041, USA
[2] Department of Computer Science and Information Systems, University of North Alabama, Florence, AL 35632, USA
[3] Department of Computer and Information Science, University of Massachusetts Dartmouth, Dartmouth, MA 02747, USA
[4] Department of Marketing, School of Business Administration, The University of Mississippi, Oxford, MS 38677, USA

Corresponding author: Yi Liu (yliu11@umassd.edu)

**ABSTRACT** Machine learning has been increasingly applied in identification of fraudulent transactions. However, most application systems detect duplicitous activities after they have already occurred, not at or near real time. Since spurious transactions are far fewer than the normal ones, the highly imbalanced data makes fraud detection very challenging and calls for ways to address it beyond the traditional machine learning approach. This study has proposed a detection framework, and implemented it using quantum machine learning (QML) approach by applying Support Vector Machine (SVM) enhanced with quantum annealing solvers. To evaluate its detection performance, we have further implemented twelve machine learning methods, and compared the performance of QML application with these machine learning implementations on two datasets: Israel credit card transactions (non-time series) which is moderately imbalanced, and a bank loan dataset (time series) that is highly imbalanced. The result shows that, the quantum enhanced SVM has categorically outperformed the rest in both speed and accuracy with the bank loan dataset. However, its detection accuracy is similar to others with Israel credit card transactions data. Furthermore, for both datasets, feature selection has been shown to significantly improve the detection speed, although the improvement on accuracy is marginal. These findings have demonstrated the potential of QML applications on time series based, highly imbalanced data, and the merit of traditional machine learning approaches in non-time series data. This study provides insight on selecting appropriate approach with different types of datasets while taking into consideration the tradeoffs of speed, accuracy, and cost.

**INDEX TERMS** Fraud detection, machine learning, quantum computing.

## I. INTRODUCTION

Fraudulent transactions are costly to businesses. According to [1], every year, businesses in the US lost 4 billion dollars on average because of fraudulent transactions, and insurance companies in the UK lost 1.6 billion pounds to the fraudulent transaction claims [2]. In addition to expense write-offs to cover shipping, refund, and other managerial expenses, businesses also lose sale opportunities from trusted customers and reputational risk [3].

Effective detection systems can assist businesses to cut down the loss due to fraudulent transactions. However, it is challenging to prevent and detect these incidents for three main reasons. First, the development and wide adoption of mobile technologies has brought a tremendous increase of online transactions. In early 2020, there has been a 110% increase in e-commerce transactions in the US alone than previous year [4], and subsequently, web attacks to e-commerce retailers and associated fraudulent activities are also on the rise [5], [6]. Secondly, in spite of the need for real time or near real time fraud detection for online e-commerce transactions, the effectiveness of many existing systems is compromised since most detect only after the fraud activities have happened

The associate editor coordinating the review of this manuscript and approving it for publication was Diego Oliva.

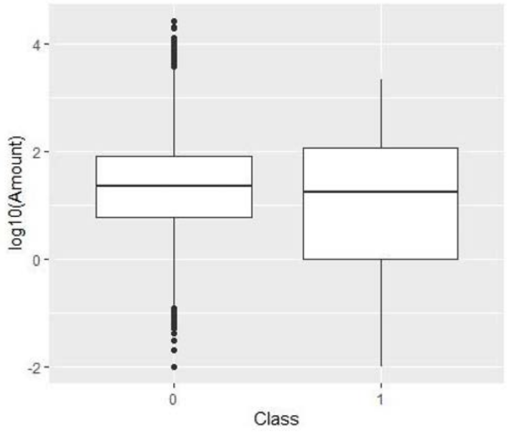


**FIGURE 1.** Distribution of amount ($ per transaction) on Normal (0) and Fraudulent (1).

when the loss has already occurred. Thirdly, since the majority of the transactions are normal and fraudulent ones are rare, the datasets are highly imbalanced with the anomalous transactions treated as outliers (anomaly data point). Figure 1 shows the box plot distribution of European cardholders' normal and fraudulent credit card transactions that occurred within two days in September 2013 [7]. From figure 1, we can see that it is not always easy to distinguish normal transactions from the fraudulent ones since the prior can have many ''outliers'' with extreme values, whereas spurious ones tend to have ''normal'' observations in terms of monetary amount per transaction, so fraudulent transactions have an either ''too weak'' or ''too diffuse'' pattern compared to the normal ones, often laborious to detect. Since the first challenge of fraud detection is the consequence of technological advancement, beyond the scope of this study, we aim to address the second and the third challenges by integrating quantum annealing solvers and machine learning algorithms to deliver quality real time/near real time fraud detection.

Online transactions produce time series data that can be classified as stationary and non-stationary. Stationary time series data remain constant over time whereas non-stationary data change over time and can behave as trends and cycles [8]. Non-stationary data tends to be unpredictable and needs to convert into stationary data for data modeling and forecast. Since non-stationary data is sensitive to the time variable, ''time'' must be included into the analytic model as an important variable. Autoregressive models are common approaches for analyzing non-stationary data: a traditional linear autoregression for the linear autocorrelation, and a deep autoregressive network with quadratic formulation. Besides the models designed for directly analyzing non-stationary data with the ''time'' variable, there are existing approaches to transform non-stationary to stationary data by detrending methods such as power transform, square root, and log transform [9].

To address the issue of response delay in fraud detection systems and imbalanced datasets, in this study, we are interested in exploring answers to the following questions:

1) How does application of quantum machine learning (QML) in fraud detection compare with traditional machine learning algorithms?
2) What is the impact of feature/variable selection on detection performance?
3) How does QML perform differently with traditional machine learning algorithms on datasets that have various characteristics, e.g., time series based verse non time series based datasets, imbalanced verse highly imbalanced datasets?

E-commerce has been consistently on the rise. According to cardrates.com, in 2021, the world wide retail e-commerce sales was approximately 4.9 trillion U.S. dollars, with 108.6 million of daily credit card transactions in the U.S.. Because of quantum computing's powerful modelling abilities to solve some complex problems that existing computing cannot, we consider QML a promising approach to tackle the huge volume of online fraud data. Details on why QML are used in this study are discussed in section II.

This study contributes to the literature on fraud detection by proposing and implementing a solution framework with QML to analyze online transaction data. Furthermore, it demonstrates the potential of QML's capability in critical business applications.

The rest of the paper is organized as follows: section II overviews the extant fraud detection literature of machine learning algorithms. It explains how Support Vector Machine (SVM) uses hyperplanes for classification, Kernel Trick - the process of converting the nonlinear support vector classifier to a linear one, and the role of quantum computing that can be applied to speed up the identification of the more complex kernel functions. Section III shows the proposed fraud detection framework and its components. Section IV describes the characteristics of the datasets used and the algorithms evaluated in this study. Section V reports the fraud detection performance comparison of SVM QUBO and other machine learning algorithms. Section VI analyzes and discusses the insight from the findings. Section VII summarizes this study and the future work.

## II. LITERATURE REVIEW ON FRAUD DETECTION AND MACHINE LEARNING ALGORITHMS

Machine learning approaches have been increasingly applied in fraud detection [20], [21]. Since the highly imbalanced data and diffused pattern affect the prediction accuracy of traditional machine learning algorithms [22], and some non-stationary data violate the assumptions of traditional clustering and classification methods, there have been increased research interests in using novice methods to tackle this problem in recent years. Although machine learning algorithms have been proposed, they are still under the assumption of stationary or non-time series data.

Table 1 summarized the typical machine learning algorithms that have been applied for fraud detection.

This study has implemented a QML system applying Support Vector Machine (SVM), a popular traditional machine

**TABLE 1. Machine learning algorithms for fraud detection.**

| Research Method | Type of Fraud | Reference |
|---|---|---|
| Neural Network | Financial report | [10] |
| Logistic Regression | Credit card transaction | [11] |
| Support Vector Machines (SVMs) | Credit card transaction, insurance, and financial report | [12] |
| Decision Tree | Credit card transaction, financial report | [13] |
| Genetic Algorithm | Credit card transaction | [14] |
| Text Mining | Financial report | [15] |
| Self-organizing Map | Credit card transaction | [16] |
| Bayesian Network | Credit card transaction | [17] |
| Artificial Immune Systems | Credit card transaction | [18] |
| Ensemble Method (SVM, KNN, NN and others) | Credit card transaction, insurance, and financial report | [19] |

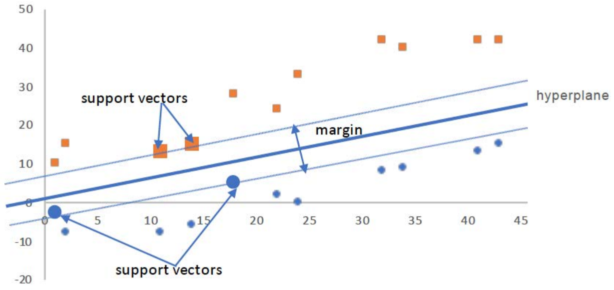
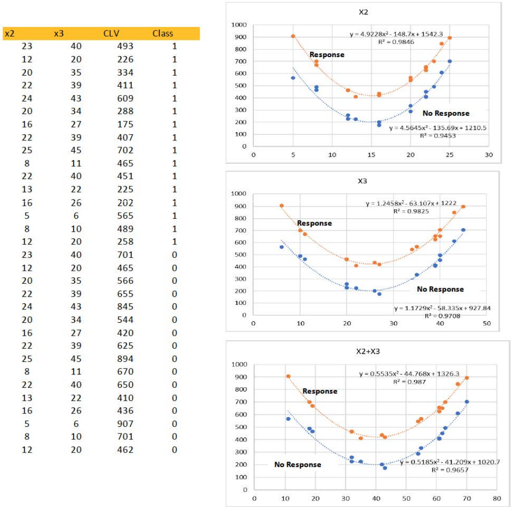


**FIGURE 2. Example of a two-group classification problem with support vectors highlighted.**

learning method, and enhanced with quantum capabilities, it then compares the performance of the system and twelve other traditional machine learning algorithms. Support Vector Machine (SVM), is a high performance, widely adopted predictive analytics method developed by Vapnik and his colleagues at AT&T Bell laboratories [23], [24]. It is a supervised machine learning method for two-group classification problems. Using linear decision functions for linear hyperplanes, SVM separates the observations into two groups by mapping the input vector into high dimensional feature space. SVM has been applied in various data analytic applications including fraud detection [12], [25]. The objective of SVM is to find a decision function that constructs the hyperplane between two groups to maximize the margin. The hyperplane that can create the maximum separating margin between the two groups is known as the optimal hyperplane, as shown in Figure 2. The training data to construct the optimal hyperplane and determine the maximum separating margin are called support vectors. Four support vectors are needed to construct the hyperplane in Figure 2.

Like other supervised learning methods, the dependent variable (classifier) must be labeled. For example, in fraud detection, the ''fraudulent status'' will be the classifier, and the transactions' characteristics the independent variables (attributes). Once the optimal hyperplane is constructed, it is then used to separate the transactions into normal and fraudulent groups. There are two types of hyperplanes: the hard margin hyperplane separates support vectors into two groups without error, and the soft one allows the minimum number of errors [26].



| x2 | x3 | CLV | Class |
|---|---|---|---|
| 23 | 40 | 493 | 1 |
| 12 | 20 | 226 | 1 |
| 20 | 35 | 334 | 1 |
| 22 | 39 | 411 | 1 |
| 24 | 43 | 609 | 1 |
| 20 | 34 | 288 | 1 |
| 16 | 27 | 175 | 1 |
| 22 | 39 | 407 | 1 |
| 25 | 45 | 702 | 1 |
| 8 | 11 | 465 | 1 |
| 22 | 40 | 451 | 1 |
| 13 | 22 | 225 | 1 |
| 16 | 26 | 202 | 1 |
| 5 | 6 | 565 | 1 |
| 8 | 10 | 489 | 1 |
| 12 | 20 | 258 | 1 |
| 23 | 40 | 701 | 0 |
| 12 | 20 | 465 | 0 |
| 20 | 35 | 566 | 0 |
| 22 | 39 | 655 | 0 |
| 24 | 43 | 845 | 0 |
| 20 | 34 | 544 | 0 |
| 16 | 27 | 420 | 0 |
| 22 | 39 | 625 | 0 |
| 25 | 45 | 894 | 0 |
| 8 | 11 | 670 | 0 |
| 22 | 40 | 650 | 0 |
| 13 | 22 | 410 | 0 |
| 16 | 26 | 436 | 0 |
| 5 | 6 | 907 | 0 |
| 8 | 10 | 701 | 0 |
| 12 | 20 | 462 | 0 |

**FIGURE 3. Example of nonlinear support vector classifier.**

In SVM, there are linear and nonlinear support vector classifiers. To facilitate the identification of the optimal hyperplane, it is necessary to transform the nonlinear support vector classifier to a linear one. Such a process is called ''Kernel Trick'' which is described below:

A linear support vector classifier has separable linear variables in the decision function (1).

$$z_l = a_1x_1 + a_2x_2 + \cdots + a_nx_n \tag{1}$$

Nonlinear support vector classifier has separable nonlinear variables in the decision function (2).

$$z_{nl} = a_1x_1^{0.5} + a_2x_2^3 + \cdots + a_nx_n^\nu \tag{2}$$

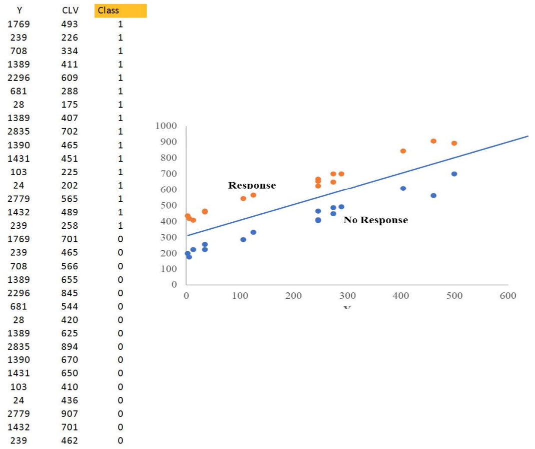
Because each term in decision function is separable, the nonlinear variables can be replaced by new linear variables:

$$y_1 = x_1^{0.5}, y_2 = x_2^3, \ldots, y_n = x_n^\nu \tag{3}$$

then the kernel trick ends with the linear classifier $z_l$ which is equivalent to

$$z_{nl} : z_l = a_1y_1 + a_2y_2 + \cdots + a_ny_n \tag{4}$$

Figures 3 and 4 come from one of the authors' teaching materials to illustrate how SVM can be applied in predicting consumer life value. In Figure 3, $X2$ and $X3$ are independent variables ($X1$ is a demographic variable thus excluded from the figure) and Consumer Life Value (CLV) a dependent variable (classifier). The support vector classifier in this example is nonlinear, and figure 4 shows how the ''kernel trick'' helps transform it into a linear one by replacing independent variables $X2$ and $X3$ with a new equation $Y = (a_1 \times 2 + b_1)^2 + (a_2 \times 3 + b_2)^2$, where $a_1 = 4.74365$, $b_1 = -71.0975$, $a_2 = 1.20935$, $b_2 = -30.3605$.



| Y | CLV | Class |
|---|---|---|
| 1769 | 493 | 1 |
| 239 | 226 | 1 |
| 708 | 334 | 1 |
| 1389 | 411 | 1 |
| 2296 | 609 | 1 |
| 681 | 288 | 1 |
| 28 | 175 | 1 |
| 1389 | 407 | 1 |
| 2835 | 702 | 1 |
| 1390 | 465 | 1 |
| 1431 | 451 | 1 |
| 103 | 225 | 1 |
| 24 | 202 | 1 |
| 2779 | 565 | 1 |
| 1432 | 489 | 1 |
| 239 | 258 | 1 |
| 1769 | 701 | 0 |
| 239 | 465 | 0 |
| 708 | 566 | 0 |
| 1389 | 655 | 0 |
| 2296 | 845 | 0 |
| 681 | 544 | 0 |
| 28 | 420 | 0 |
| 1389 | 625 | 0 |
| 2835 | 894 | 0 |
| 1390 | 670 | 0 |
| 1431 | 650 | 0 |
| 103 | 410 | 0 |
| 24 | 436 | 0 |
| 2779 | 907 | 0 |
| 1432 | 701 | 0 |
| 239 | 462 | 0 |

**FIGURE 4.** Example of kernel trick for nonlinear support vector classifier.

Even with moderate size of data on nonlinear classifiers, the process of constructing kernel functions in SVM is time consuming. More complex kernel functions can be obtained by solving quadratic constrained binary optimization problem [27], and that require very high computing capability. One solution is to develop a general quadratic constrained model for SVM, and recast it explicitly as a quadratic unconstrained binary optimization problem (QUBO) using quadratic infeasibility penalties as constraints [28]. Since the problems need to be converted into a QUBO format, the difficulty of this conversion process has made it one of the bottlenecks for the wide applications of quantum computing. Quantum computing has experienced some level of success in solving the specific application in QUBO formulation [29]. The successful implementation experiments of such a solution [30], [31] are very encouraging, motivating us to explore its applications in fraud detection.

There are technological difficulties as well as practical issues for transforming quadratic constrained binary optimization problems into QUBO. In addition, there is a lack of benchmarks to compare the performance of quantum computing and traditional computing. Without demonstrating superlative results, given how costly quantum computing is, it is challenging to promote it to a broader user base. Thus, technology acceleration on quantum computing itself might not mean wide application and adoption if there are no economies of scale and network effect due to limited users.

Another bottle neck for quantum computing is the tremendous effort required to redesign the existing algorithms and data structures built on traditional computing platforms. Because of the cost and effort of using quantum computing, it is crucial to apply it only on important applications. Fraud detection for online transactions is a perfect one for such purpose.

## III. EMPIRICAL FRAMEWORK AND METHODOLOGIES

We propose a fraud detection framework as shown in Figure 5. The framework first verifies whether the input data is time series-based vs static, followed by a stationary test to determine whether the time series data are stationary or non-stationary. Since Augmented Dickey Fuller (ADF) and Kwiatkowski-Phillips-Schmidt-Shin (KPSS) are two of the most commonly used statistical test to analyze whether the series of data are the stationary, this study uses both tests [32], [33] to evaluate whether the time series data is stationary as shown in Figure 5 via the unit root test. For non-stationary data, several common detrending methods such as power transform, square root, and log transform, will be applied to convert them into stationary. Then the dimension reduction technique is used to reduce the "noise" attributes of the data.

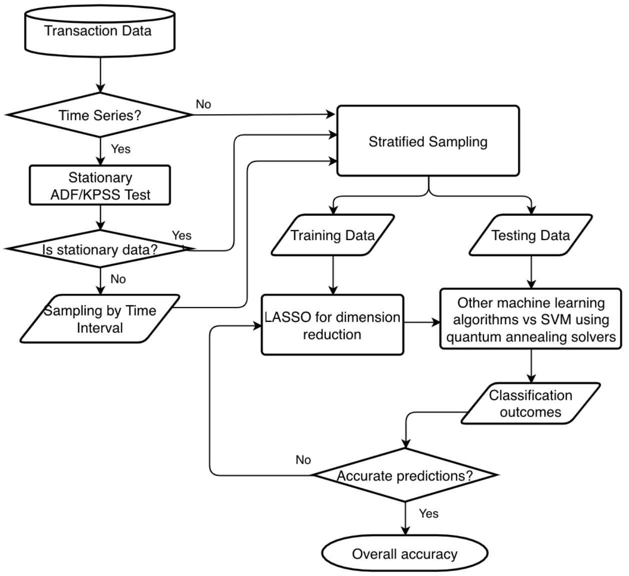


**FIGURE 5.** Fraud detection framework.

To eliminate variables that do not contribute to the prediction accuracy or are "noises" that reduce it, we use Least Absolute Shrinkage and Selection Operator (LASSO) [34] to construct better prediction models. Given a set of linear independent variables (features or attributes), the estimator for the predictor (linear classifier) $y$ is:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n \tag{5}$$

LASSO function is defined as:

$$\begin{aligned} &\min \Sigma(y - \hat{y})^2 \\ &st.\Sigma|\beta_i| \leq s, where\ i = 1, \ldots, n \end{aligned} \tag{6}$$

To remove inconsequential features (vectors) from the model, we can decrease the value of $s$ to force some $\beta_i$ to be 0, thus drop some non contributing independent variables (attributes) to improve fraud detection.

The machine learning approach of obtaining kernel functions of SVM will be formulated as QUBO, and then the kernel functions identified by quantum annealing solvers will be applied on predictive analysis of fraud detection. The performance of this QML fraud detection system will then be

**TABLE 2.** Imbalanced data with fraudulent transactions.

| instance | total # of transactions | Time series | # of independent variables (attributes) | Positive Class (# of Fraudulent Transactions) |
|---|---|---|---|---|
| ICCT | 14,999 | No | 29 | 3,571 |
| LOAN | 33,320 | Yes | 122 | 265 |

**TABLE 3.** Machine learning algorithms used.

| type | Machine learning algorithms |
|---|---|
| Unsupervised | Copula-Based Outlier Detection — COPOD [37]<br>K-Nearest Neighbor (KNN) [38]<br>Restricted Boltzmann Machine (RBM) [39] |
| Supervised | Balanced Bagging [40]<br>Balanced Random Forest [41]<br>Linear Discriminant Analysis (LDA) [42]<br>Logistic Regression (LR) [43]<br>Logistic Regression - balanced [44]<br>Neural Network - MLP [45]<br>Random Forest (RF) [46]<br>Random Forest (RF) - balanced [47]<br>Ensemble Method - Regression Trees (RT) [48] and Logistic Regression(LR) |

compared to the system built with other traditional machine learning algorithms. In this study, the benchmarks we use are the speed and prediction accuracy of twelve traditional machine learning methods (see Table 4), and ROC curve to measure false positive. The next session will provide more details.

## IV. RESEARCH DESIGN

The rise of fraud incidents makes it important to learn more about characteristics of datasets associated with different types of frauds. Such understanding will help to implement and better identify system for fraud detection. For this purpose, this study has selected two datasets: Israeli cardholders' credit cards transactions (ICCT) [35] and bank loans application data [36] (see Table 2). The ICCT dataset is non-time-series based, contains 14,999 transactions with 23.8% fraudulent cases (3,571 in total) and 29 independent variables. For every 100 transactions in this dataset, there are more than 23 fraud cases. The LOAN dataset is time-series based, having 33,320 transactions with 0.798% fraudulent transactions (265 cases) and 122 independent variables. Compared to ICCT dataset, in additional to being highly dimensional, the LOAN data is also highly imbalanced with only less than 1 fraud case for every 1000 cases.

SVM-QUBO and twelve traditional machine learning algorithms are applied to both datasets to predict whether a transaction is a fraud (1) or not (0), and then their performances in speed and prediction accuracy are compared.

Table 3 lists the twelve traditional machine learning algorithms. Three are unsupervised learning methods and the rest are supervised ones. The difference between *Logistic Regression* and *Logistic Regression - balanced* is the setting of the *class_weight* while using the Scikit Learn library [44]: the former sets the *class_weight* to be "none" and the latter sets it to "balanced". The difference between *Random Forest* and *Random Forest - balanced* is similar.

In each dataset, 67% of data is used as the training set and the remaining as the testing set. The data processing and solution framework including ADF/KPSS test, LASSO, and the machine learning methods are coded in Python 3.8. The execution environment for the traditional machine learning methods is a PC with Microsoft Windows 10, AMD Dual-Core Gold 3150U 2.4GHz with 16GB RAM. Since D-Wave is open source, powerful, and a leader in quantum computing with a fast growing user base, we choose D-Wave quantum annealing solvers such as *dwave-hybrid* [49] to solve the QUBO model of SVM in this study.

## V. RESULTS

Both speed and accuracy comparisons of SVM-QUBO and the twelve machine learning algorithms are conducted on each dataset (Table 4, 5, 6, 7). In terms of speed, the entire execution time for SVM-QUBO includes time for (1) I/O to create the training and testing files and folders, and (2) training and testing the model. Since other traditional methods do not need to create folders, for direct comparison of speed performance, only training and testing time are used. Overall, the evaluation results show that the SVM-QUBO delivers the fastest speed in detection, outperforming traditional machine learning algorithms in speed and accuracy with loan dataset which is time-series based, highly dimensional, and highly imbalanced; whereas traditional machine learning algorithms is a better choice for ICCT data which is non time-series based and moderately imbalanced. Although feature selection significantly shortens the speed for majority of the algorithms, its contribution to improve detection accuracy is not obvious. Details are provided next.

False positive refers to incorrectly identified the normal transactions as fraudulent ones. In business, the cost of a false positive often out weights a false negative. When a legitimate customer is misidentified as a fraud, the negative experience could lead to the loss of that customer [50]. To further evaluate and compare the performance of QML and other machine learning algorithms, in this study, we examine the false positive rate using AUC(Area Under Curve) ROC(Receiver optimization Characteristics) Curve. The evaluation results will be elaborated in the next two sections.

### A. EVALUATION RESULTS: ICCT DATASET

Tables 4 and 5 report the comparison results of SVM-QUBO versus the twelve machine learning algorithms on the testing set of ICCT dataset, with and without feature selection.

As shown in Table 4, when no feature selection method is applied, SVM-QUBO is 4.4 times faster than the fastest traditional algorithm (Linear Discriminant Analysis), 34 times faster than the median, and 3,554 times faster than the one taking the longest time (Restricted Boltzmann Machine). In terms of overall accuracy, 8 out of 12 traditional algorithms have better overall accuracy than SVM-QUBO, although the lead is marginal with SVM-QUBO performing just 1.41% lower than the most accurate one - Random Forest(balanced).

**TABLE 4. SVM-QUBO verse machine learning algorithms on ICCT dataset with no feature selection.**

| Method | Time in seconds | False Negative /4950 | False Positive /4950 | Correct prediction /4950 | Overall Accuracy (10 folds) |
|---|---|---|---|---|---|
| SVM-QUBO | 0.07146 | 90 | 41 | 4819 | 0.97354 |
| Balanced Bagging | 2.70452 | 49 | 19 | 4882 | 0.98577 |
| Balanced RF | 2.20419 | 28 | 22 | 4900 | 0.98726 |
| LDA | 0.31319 | 232 | 6 | 4712 | 0.95363 |
| LR | 1.29326 | 47 | 9 | 4894 | 0.98517 |
| LR - balanced | 1.27573 | 36 | 36 | 4878 | 0.98398 |
| NN - MLP | 4.11368 | 39 | 15 | 4896 | 0.98517 |
| RF | 5.72131 | 41 | 5 | 4904 | 0.98746 |
| RF - balanced | 4.57544 | 35 | 5 | 4910 | 0.98766 |
| Ensemble: RT- LR | 1.58578 | 28 | 54 | 4868 | 0.9796 |
| COPOD | 1.10148 | 736 | 34 | 4180 | 0.84785 |
| KNN | 5.24852 | 893 | 169 | 3888 | 0.79302 |
| RBM | 254.00183 | 2 | 3730 | 1218 | 0.24211 |

**TABLE 5. SVM-QUBO verse machine learning algorithms on ICCT dataset with LASSO for feature selection.**

| Method | Time in seconds | False Negative /4950 | False Positive /4950 | Correct prediction /4950 | Overall Accuracy (10 folds) |
|---|---|---|---|---|---|
| SVM-QUBO | 0.02814 | 89 | 13 | 4848 | 0.97939 |
| Balanced Bagging | 1.32265 | 52 | 20 | 4878 | 0.98418 |
| Balanced RF | 1.61032 | 30 | 30 | 4890 | 0.98607 |
| LDA | 0.22967 | 244 | 5 | 4701 | 0.94915 |
| LR | 0.23767 | 56 | 15 | 4879 | 0.98209 |
| LR - balanced | 0.25117 | 49 | 49 | 4852 | 0.979 |
| NN - MLP | 2.28024 | 47 | 29 | 4874 | 0.98298 |
| RF | 3.57253 | 50 | 6 | 4894 | 0.98637 |
| RF - balanced | 2.72568 | 47 | 6 | 4897 | 0.98667 |
| Ensemble: RT- LR | 1.6217 | 41 | 61 | 4848 | 0.9799 |
| COPOD | 0.62624 | 715 | 16 | 4219 | 0.85909 |
| KNN | 2.60529 | 911 | 173 | 3866 | 0.78237 |
| RBM | 173.70629 | 29 | 1527 | 3394 | 0.67897 |

When LASSO is applied for feature selection, it further shortens the execution time for SVM-QUBO. As shown in Table 5, SVM-QUBO is 8 times faster than the fastest traditional algorithm (Linear Discriminant Analysis), 57 times faster than the median, and 6173 times faster than the one taking the longest time (Restricted Boltzmann Machine). However, there is no obvious performance gain in accuracy by applying LASSO for feature selection than without. Similar to results without feature selection, 7 out of 12 traditional algorithms have slightly better overall accuracy than SVM-QUBO, whereas SVM-QUBO is 0.728% lower than the highest accurate one - Random Forest(balanced).

The comparison of the speed performance shows that applying LASSO significantly reduces the execution time, with saving 27% for the lowest and 82% the highest (LR-balanced). One exception is the Ensemble RT-LR, which takes 2.2% longer with LASSO than without feature selection.

The AUROC curve shown in Figure 6 plots the true positive rate against the false positive rate [51], showing that for ICCT dataset, SVM-QUBO performs the best with the value of 0.99, closely followed by supervised machine learning methods. The unsupervised learning such as COPOD, KNN, and RBM do not perform as well, among which RBM performs the best with the value of 0.78 when feature selection is applied.

**TABLE 6. SVM-QUBO verse machine learning algorithms on LOAN dataset with no feature selection.**

| Method | Time in seconds | False Negative /10996 | False Positive /10996 | Correct prediction /10996 | Overall Accuracy (10 folds) |
|---|---|---|---|---|---|
| SVM-QUBO | 0.09263 | 760 | 52 | 10184 | 0.92615 |
| Balanced Bagging | 3.72162 | 742 | 146 | 10108 | 0.86413 |
| Balanced RF | 1.76025 | 331 | 3583 | 7082 | 0.63249 |
| LDA | 0.51514 | 752 | 34 | 10210 | 0.87088 |
| LR | 1.01368 | 763 | 0 | 10233 | 0.87218 |
| LR - balanced | 0.46915 | 347 | 4297 | 6352 | 0.5789 |
| NN - MLP | 0.45193 | 763 | 0 | 10233 | 0.87231 |
| RF | 4.21333 | 761 | 3 | 10232 | 0.87179 |
| RF - balanced | 3.9576 | 763 | 1 | 10232 | 0.87243 |
| Ensemble: RT- LR | 3.85063 | 388 | 3238 | 7370 | 0.6456 |
| COPOD | 2.32763 | 710 | 1078 | 9208 | 0.79081 |
| KNN | 10.72006 | 720 | 950 | 9326 | 0.78926 |
| RBM | 260.5561 | 763 | 0 | 10233 | 0.87218 |

**TABLE 7. SVM-QUBO verse machine learning algorithms on LOAN dataset with LASSO for feature selection.**

| Method | Time in seconds | False Negative /10996 | False Positive /10996 | Correct prediction /10996 | Overall Accuracy (10 folds) |
|---|---|---|---|---|---|
| SVM-QUBO | 0.06601 | 762 | 63 | 10171 | 0.92497 |
| Balanced Bagging | 0.76276 | 754 | 86 | 10156 | 0.86504 |
| Balanced RF | 1.3683 | 332 | 4468 | 6196 | 0.55295 |
| LDA | 0.25688 | 763 | 0 | 10233 | 0.87218 |
| LR | 0.45304 | 763 | 0 | 10233 | 0.87218 |
| LR - balanced | 0.36939 | 349 | 4143 | 6504 | 0.57552 |
| NN - MLP | 0.31607 | 0 | 10233 | 763 | 0.12782 |
| RF | 2.3978 | 760 | 11 | 10225 | 0.87062 |
| RF - balanced | 2.39944 | 761 | 2 | 10233 | 0.87218 |
| Ensemble: RT- LR | 2.89756 | 448 | 3427 | 7121 | 0.63769 |
| COPOD | 0.40503 | 728 | 1086 | 9182 | 0.78705 |
| KNN | 2.26045 | 697 | 1026 | 9273 | 0.79419 |
| RBM | 184.08431 | 763 | 0 | 10233 | 0.87218 |

### *B. EVALUATION RESULTS: LOAN DATASET*

Tables 6 and 7 report the comparisons of applying SVM-QUBO versus the twelve machine learning algorithms on the testing set of LOAN dataset, with no feature selection and LASSO applied. SVM-QUBO significantly outperforms all the machine learning algorithms both in speed and overall accuracy regardless whether feature selection is applied.

In terms of speed, when no feature selection method is applied, SVM-QUBO is 5 times faster than the fastest machine learning (Logistic Regression - balanced), 32 times faster than the median, and 2813 times faster than Restricted Boltzmann Machine, the one takes the longest time. When LASSO is applied, SVM-QUBO is 3.8 times faster than the fastest machine learning (Linear Discriminant Analysis),

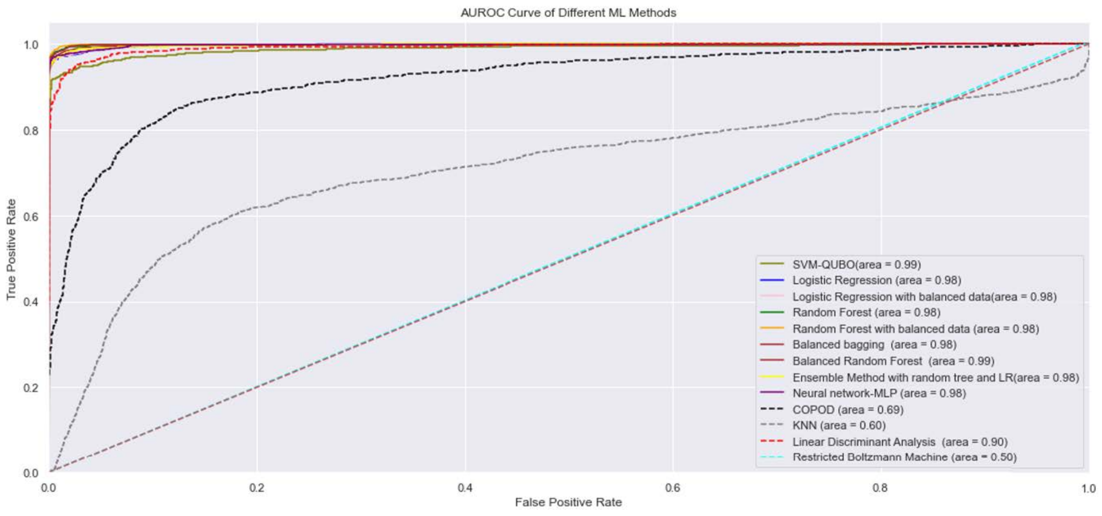


**FIGURE 6.** AUROC curves of SVM-QUBO verses machine learning algorithms on ICCT dataset with no feature selection.

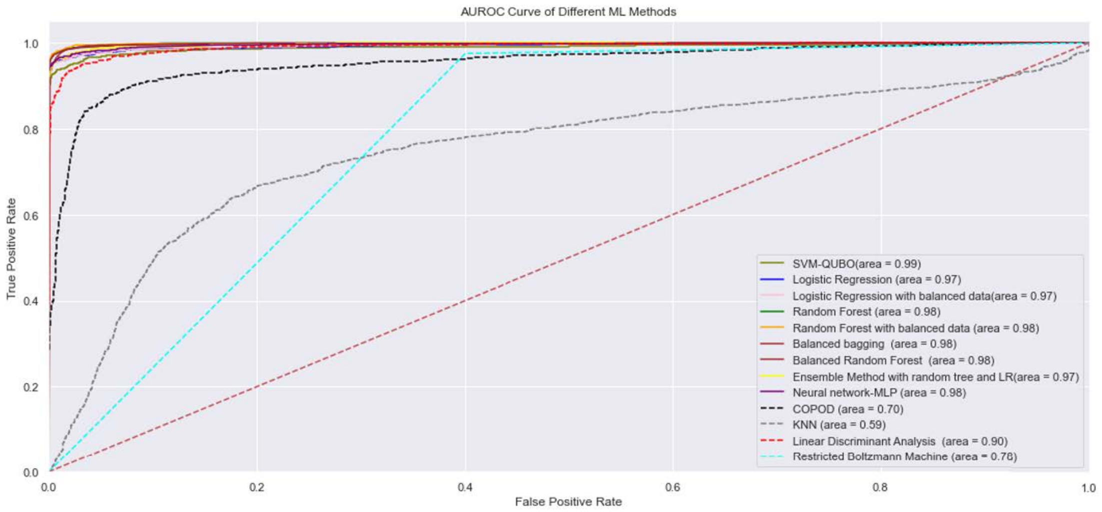


**FIGURE 7.** AUROC curves of SVM-QUBO verses machine learning algorithms on ICCT dataset with LASSO.

16 times faster than the median, and 2788 times faster than the one taking the longest time (Restricted Boltzmann Machine).

SVM-QUBO has 5.3% higher overall accuracy than the best performed traditional machine learning algorithm (Random Forest-balanced) without feature selection, and 6.2% higher with feature selection than the best performed traditional ones, including Linear Discriminant Analysis, Logistic Regression, Random Forest-balanced and Restricted Boltzmann Machine, with LASSO.

The AUROC curves of SVM-QUBO and the other machine learning algorithms with and without LASSO are shown in Figure 8 and 9. The AUROC curve shows that overall, none of these algorithms perform well. Algorithm that performs the best is Balanced Random Forest with an area of 0.61 for LOAN dataset without LASSO feature selection, and logistic Regression with area of 0.57 with LASSO feature selection. SVM-QUBO performs slightly better than most, yet it is still as low as 0.57 with no feature selection, and 0.51 with feature selection.

All machine learning algorithms gain significantly in speed performance with LASSO than without, reducing 21% execution time as the lowest (Logistic Regression - balanced) and 83% as the highest (COPOD).

## VI. DISCUSSION

Findings of this study show that, QML is the fastest than all other algorithms. In terms of accuracy, for ICCT dataset which is moderately imbalanced with 23.8% fraudulent rate, QML is not an optimal option both economically and performance wise. The accuracy of QML without feature selection is worse than 9 out of 12 of the machine learning but is improved significantly with feature selection (better than 9 out of 12). The accuracy differences in performance with and without feature selection are mostly

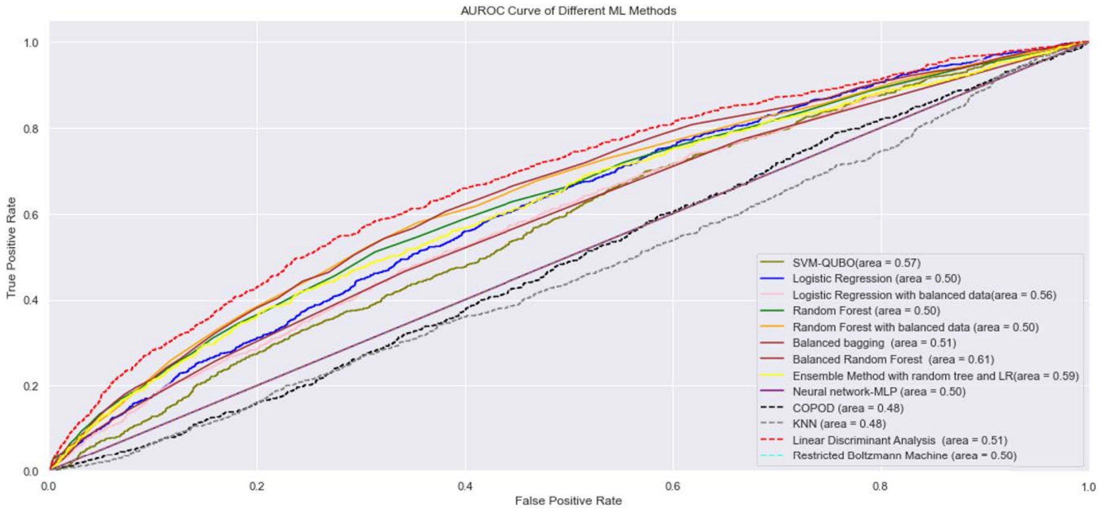


**FIGURE 8.** AUROC curves of SVM-QUBO verses machine learning algorithms on LOAN dataset without feature selection.

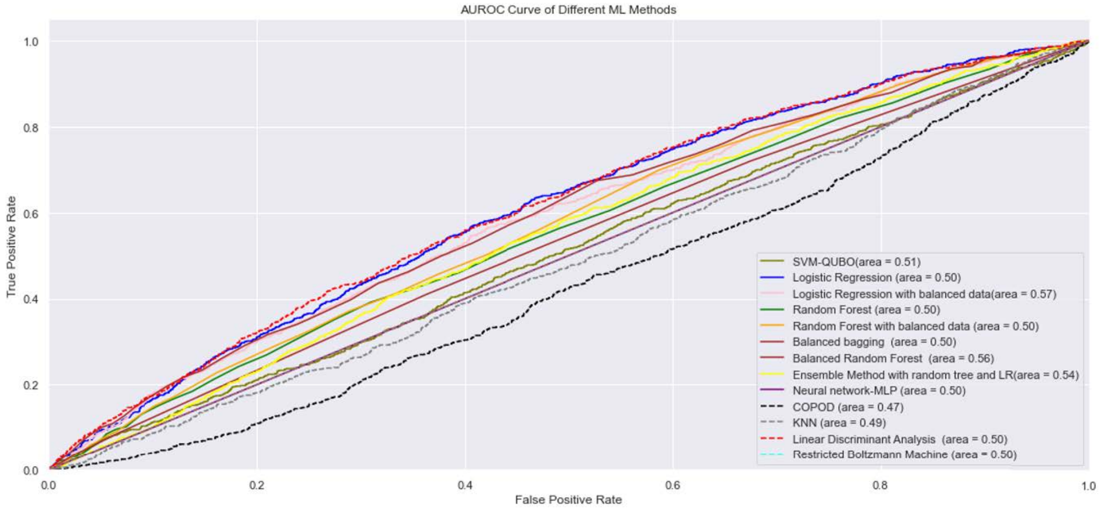


**FIGURE 9.** AUROC curves of SVM-QUBO verses machine learning algorithms on LOAN dataset with LASSO.

around 0.01%. Unsupervised machine learning algorithms do not compare well with the supervised ones, the inclusion of feature selection makes a big difference to improve the accuracy of certain unsupervised learning algorithm, for example, RBM increases 2.8 times with LASSO than without, making its performance in accuracy comparable to those of supervised machine learning algorithms.

Comparing to ICCT, for LOAN dataset that is time-series based, non-stationary, and highly imbalanced, QML performs categorically better than the other machine learning algorithms in terms of speed and overall accuracy, regardless of whether feature selection is included. We also notice that unsupervised learning RBM performs better than other supervised learning for the LOAN dataset.

In terms of using AUROC curve to compare the false positive of these algorithms, QML and machine learning algorithms perform well with ICCT dataset. SVM QUBO has AUROC score as high as 0.99, and the 9 supervised machine learning algorithms have AUROC scores as high as 0.98. However, the AUROC score for LOAN data shows a different picture: all algorithms including QML hovering around 0.5 values, only one with 0.60. This suggest how difficulties it is to process highly imbalanced and high dimensional dataset.

Based on evaluation results from this study, to summarize, we find that given how the costly quantum computing is, before quantum hardware has major improvement, traditional machine learning methods might be a good solution to deliver satisfactory results for moderately imbalanced, non-time-series data, whereas as for highly imbalanced, high dimensional, time-series based data, it would be worthwhile to consider QML. Of course such recommendations warrant more tests on different types of data prior making it generalized recommendation.

This study is one of the few QML applications in fraud detection, a worthwhile endeavor to expand the problems that can take advantage of quantum computing's capabilities. Also, its performance comparison with as many as twelve other machine learning algorithms with different characteristics (supervised and unsupervised) make this study comprehensive and unique.

## VII. CONCLUSION

QML has drawn an increasing interest for its potential of solving critical problems because of its computing capability [9], [20]. Due to the challenges of formulating problems into the QUBO format that quantum computing requires, and how costly such a process is, identification of the most critical areas for QML application is important to justify the expense for potential gain in performance. The prevalence of e-business, online transactions, and huge loss over spurious activities, make timely and accurate fraud detection a great choice for QML solution. We propose a fraud detection framework for this purpose. This framework will first determine whether the data is time-series based, and then whether the data is stationary or non-stationary, next it will apply feature selection to eliminate the ''noises'' from the data prior fraud detection. To evaluate the effectiveness of the proposed framework, we implement a QML system using SVM enhanced with quantum annealing solver, and compare its performance in detection speed, accuracy, and false positive rate with 12 other traditional machine learning algorithms on both datasets (time-series based, highly imbalanced, high dimensional verse non-time-series-based, moderate imbalanced).

This study compares the performance of a QML system with those built on 12 machine learning algorithms on fraud detection. The findings show the effectiveness of our proposed fraud detection framework, and the outstanding performance of QML, specifically with a time-series based, highly imbalanced, high dimensional dataset. Our study contributes to the detection literature by serving as a road map for the further research in QML. Despite the growing interest in applying quantum to solve real-world problems, our results caution that, to provide the most optimum business solution, the practitioners need to take the tradeoffs of accuracy, speed, and the cost of computing into consideration, as well as the kind of data the system works with. Quantum computing applications can have great potential in processing datasets characterized with time-series, high dimensional and highly imbalanced, which are challenging for traditional machine learning algorithms. As for the non time-series data, before quantum computing makes significant breakthrough, the traditional machine learning still plays an important role for being the more economic and effective solution.

Given the extremely fast detection speed, the QML system in this study delivers near real time results, which is a very important step towards the direction of real time fraud detection. As a continuous study, we are developing a simulator to generate datasets with different numbers of transactions and different ratios between normal and fraudulent transactions. The synthesized datasets use the similar settings from the benchmark instances used in this study. Unlike the studies in the literature that conduct analysis of the transaction data after the transactions occurred, it will create unique experiments to generate live transaction data and real time fraud detection through this simulator.

## REFERENCES

[1] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, ''Data mining for credit card fraud: A comparative study,'' *Decis. Support Syst.*, vol. 50, no. 3, pp. 602–613, Feb. 2011.

[2] E. W. T. Ngai, Y. Hu, Y. H. Wong, Y. Chen, and X. Sun, ''The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature,'' *Decis. Support Syst.*, vol. 50, no. 3, pp. 559–569, Feb. 2011.

[3] J. A. Tackett, ''Association rules for fraud detection,'' *J. Corporate Accounting Finance*, vol. 24, no. 4, pp. 15–22, May 2013.

[4] L. Columbus. (2020). *How E-Commerce's Explosive Growth is Attracting Fraud*. [Online]. Available: https://www.forbes.com/sites/louiscolumbus/2020/05/18/how-e-commerces-explosive-growth-is-attracting-fraud/?sh=5a1e8f586c4b

[5] LexisNexis. (2020). *Lexis/Nexis Solutions for 2020 True Cost of Fraud Study: E-Commerce/Retail Edition*. [Online]. Available: https://risk.lexisnexis.com/about-us/press-room/press-release/20200721-tcof-retail-study

[6] I.-C. Yeh and C.-H. Lien, ''The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients,'' *Expert Syst. Appl.*, vol. 36, pp. 2473–2480, Mar. 2009.

[7] Kaggle.com. *Credit Card Fraud Detection*. Accessed: May 2022. [Online]. Available: https://www.kaggle.com/mlg-ulb/creditcardfraud

[8] G. P. Nason, *Statistics in Volcanology* (Stationary and Non-Stationary Time Series). London, U.K.: Geological Society of London, 2006.

[9] R. Salles, K. Belloze, F. Porto, P. H. Gonzalez, and E. Ogasawara, ''Nonstationary time series transformation methods: An experimental review,'' *Knowl.-Based Syst.*, vol. 164, pp. 274–291, Jan. 2019.

[10] P. Ravisankar, V. Ravi, G. R. Rao, and I. Bose, ''Detection of financial statement fraud and feature selection using data mining techniques,'' *Decision Support Syst.*, vol. 50, no. 2, pp. 491–500, 2011.

[11] F. Itoo, Meenakshi, and S. Singh, ''Comparison and analysis of logistic regression, Naïve Bayes and KNN machine learning algorithms for credit card fraud detection,'' *Int. J. Inf. Technol.*, vol. 13, pp. 1503–1511, Feb. 2021.

[12] N. Rtayli and N. Enneya, ''Enhanced credit card fraud detection based on SVM-recursive feature elimination and hyper-parameters optimization,'' *J. Inf. Secur. Appl.*, vol. 55, Dec. 2020, Art. no. 102596.

[13] C. Lin, A. Chiu, S. Y. Huang, and D. C. Yen, ''Detecting the financial statement fraud: The analysis of the differences between data mining techniques and experts' judgments,'' *Knowl.-Based Syst.*, vol. 89, pp. 459–470, Nov. 2015.

[14] E. Duman and M. H. Ozcelik, ''Detecting credit card fraud by genetic algorithm and scatter search,'' *Expert Syst. Appl.*, vol. 38, no. 10, pp. 13057–13063, Sep. 2011.

[15] P. Hájek and R. Henriques, ''Mining corporate annual reports for intelligent detection of financial statement fraud—A comparative study of machine learning methods,'' *Knowl.-Based Syst.*, vol. 128, pp. 139–152, Jul. 2017.

[16] D. Olszewski, ''Fraud detection using self-organizing map visualizing the user profiles,'' *Knowl.-Based Syst.*, vol. 70, pp. 324–334, Nov. 2014.

[17] A. G. C. de Sá, A. C. M. Pereira, and G. L. Pappa, ''A customized classification algorithm for credit card fraud detection,'' *Eng. Appl. Artif. Intell.*, vol. 72, pp. 21–29, Jun. 2018.

[18] N. S. Halvaiee and M. K. Akbari, ''A novel model for credit card fraud detection using artificial immune systems,'' *Appl. Soft Comput.*, vol. 24, pp. 40–49, Nov. 2014.

[19] E. Kim, J. Lee, H. Shin, H. Yang, S. Cho, S.-K. Nam, Y. Song, J.-A. Yoon, and J.-I. Kim, ''Champion-challenger analysis for credit card fraud detection: Hybrid ensemble and deep learning,'' *Expert Syst. Appl.*, vol. 128, pp. 214–224, Aug. 2019.

[20] S. V. S. S. Lakshmi and S. D. Kavilla, ''Machine learning for credit card fraud detection system,'' *Int. J. Appl. Eng. Res.*, vol. 13, no. 24, pp. 16819–16824, 2018.

[21] R. Sailusha, V. Gnaneswar, R. Ramesh, and G. R. Rao, "Credit card fraud detection using machine learning," in *Proc. 4th Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, May 2020, pp. 1264–1270.

[22] F. Thabtah, S. Hammoud, F. Kamalov, and A. Gonsalves, "Data imbalance in classification: Experimental evaluation," *Inf. Sci.*, vol. 513, pp. 429–441, Mar. 2020.

[23] V. Vapnik, *Estimation of Dependences Based on Empirical Data*, 2nd ed. New York, NY, USA: Springer-Verlag, 2006.

[24] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[25] N. K. Gyamfi and J.-D. Abdulai, "Bank fraud detection using support vector machine," in *Proc. IEEE 9th Annu. Inf. Technol., Electron. Mobile Commun. Conf. (IEMCON)*, Nov. 2018, pp. 37–41.

[26] E. Bingham, *Advances in Independent Component Analysis and Learning Machines*. New York, NY, USA: Academic, 2015.

[27] M. C. Ferris and T. S. Munson, "Interior-point methods for massive support vector machines," *SIAM J. Optim.*, vol. 13, no. 3, pp. 783–804, Jan. 2002.

[28] G. A. Kochenberger, F. Glover, and H. Wang, "Binary unconstrained quadratic optimization problem," in *Handbook of Combinatorial Optimization*, P. M. Pardalos, D.-Z. Du, and R. L. Graham, Eds. New York, NY, USA: Springer, 2013, pp. 533–557.

[29] J. Li and S. Ghosh, "Quantum-soft QUBO suppression for accurate object detection," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 158–173.

[30] P. Date, D. Arthur, and L. Pusey-Nazzaro, "QUBO formulations for training machine learning models," *Sci. Rep.*, vol. 11, no. 1, p. 10029, Dec. 2021.

[31] D. Willsch, M. Willsch, H. De Raedt, and K. Michielsen, "Support vector machines on the D-wave quantum annealer," *Comput. Phys. Commun.*, vol. 248, Mar. 2020, Art. no. 107006.

[32] D. Dickey and W. Fuller, "Distribution of the estimators for autoregressive time series with a unit root," *J. Amer. Stat. Assoc.*, vol. 74, pp. 427–431, Jun. 1979.

[33] D. Kwiatkowski, P. C. B. Phillips, P. Schmidt, and Y. Shin, "Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root?" *J. Econometrics*, vol. 54, nos. 1–3, pp. 159–178, 1992.

[34] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc., B (Methodol.)*, vol. 58, no. 1, pp. 267–288, 1996.

[35] R. Polantizer. (2021). *Fraud Detection in Python; Predict Fraudulent Credit Card Transactions*. [Online]. Available: https://medium.com/@polanitzer/fraud-detection-in-python-predict-fraudulent-credit-card-transactions-73992335dd90

[36] Mishra5001. (2019). *Credit Card Fraud Detection: Explore All Possibilities While Sanctioning a Loan to Any Customer*. [Online]. Available: https://www.kaggle.com/mishra5001/credit-card

[37] Z. Li, Y. Zhao, N. Botta, C. Ionescu, and X. Hu, "COPOD: Copula-based outlier detection," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2020, pp. 1118–1123.

[38] N. S. Altman, "An introduction to kernel and nearest-neighbor non-parametric regression," *Amer. Statistician*, vol. 46, no. 3, pp. 175–185, Aug. 1992.

[39] L.-W. Kim, S. Asaad, and R. Linsker, "A fully pipelined FPGA architecture of a factored restricted Boltzmann machine artificial neural network," *ACM Trans. Reconfigurable Technol. Syst.*, vol. 7, no. 1, pp. 1–23, Feb. 2014.

[40] A. Lazarevic and V. Kumar, "Feature bagging for outlier detection," in *Proc. 11th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2005, pp. 157–166.

[41] Imbalanced Learn. *Balanced Random Forest Classifier*. Accessed: May 2022. [Online]. Available: https://imbalanced-learn.org/stable/references/generated/imblearn.ensemble.BalancedRandomForestClassifier.html

[42] A. J. Izenman, "Linear discriminant analysis," in *Modern Multivariate Statistical Techniques* (Springer Texts in Statistics). New York, NY, USA: Springer, 2013.

[43] W. Chen, Y. Chen, Y. Mao, and B. Guo, "Density-based logistic regression," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2013.

[44] Scikit Learn. *Logistic Regression*. Accessed: May 2022. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[45] F. Murtagh, "Multilayer perceptrons for classification and regression," *Neurocomputing*, vol. 2, nos. 5–6, pp. 183–197, Jul. 1991.

[46] M. Pal, "Random forest classifier for remote sensing classification," *Int. J. Remote Sens.*, vol. 26, no. 1, pp. 217–222, Jan. 2005.

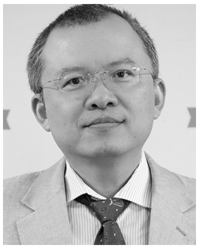
[47] Scikit Learn. *Random Forest Classifier*. Accessed: May 2022. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[48] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Monterey, CA, USA: Brooks/Cole, 1984.

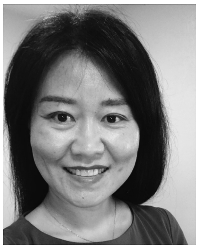
[49] D-Wave. *Dwave-Hybrid*. Accessed: May 2022. [Online]. Available: https://docs.ocean.dwavesys.com/en/stable/docs_hybrid/sdk_index.html

[50] S. Akbar and S. K. Saritha, "QML based community detection in the realm of social network analysis," in *Proc. 11th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2020, pp. 1–7.

[51] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognit.*, vol. 30, no. 7, pp. 1145–1159, 1997.



**HAIBO WANG** received the B.S. degree in biochemical engineering from the South China University of Technology, China, in 1991, and the M.S. degree in chemistry, the M.S. degree in computer science, and the Ph.D. degree in business administration from The University of Mississippi, Oxford, USA, in 1996, 1997, and 2004, respectively. He is currently a Radcliff Killam Distinguished Professor in decision science and operations research with Texas A&M International University, Laredo, TX, USA. He has publications in such outlets as IEEE TRANSACTIONS journals, *Omega*, *EJOR*, and other major OR journals. His current research interests include prescriptive analytics of big data in logistics, public transportation planning, information security, and health care.



**WENDY WANG** received the Ph.D. degree in management information systems from The University of Mississippi, in 2002. She has published book chapters and manuscripts in the *Journal of the American Society for Information Science and Technology*, *International Journal of Enterprise Information Systems*, and *Journal of Decision Systems*. Her research interest includes the behavioral and social aspects of technology.

**YI LIU** received the Ph.D. degree in computer science from The University of Mississippi, in 2005. She is currently an Associate Professor with the Department of Computer and Information Science, University of Massachusetts Dartmouth, MA, USA. Her research interests include software frameworks, software design patterns, software engineering aspects of cybersecurity, and geospatial data science. She received NIH and NASA grants for her research on using remote sensing datasets to detect and forecast outbreaks of mosquito-borne diseases.

**BAHRAM ALIDAEE** received the B.S. degree in business administration from the University of Tehran, Iran, in 1975, the M.B.A. degree from the University of North Texas, USA, in 1981, and the Ph.D. degree in mathematical sciences from the University of Texas at Arlington, in 1988. He is currently a Professor in operations and supply chain with the Business School, The University of Mississippi. He has published in *Management Science*, *Production and Operations Management*, *Transportation Science*, various IEEE TRANSACTIONS, and other major operations journals. His research interest includes applied management science interests of businesses.