Applications of Quantum Machine Learning for Quantitative Finance

PIOTR MIRONOWICZ, Department of Algorithms and Systems Modelling, Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, Poland, Department of Physics, Stockholm University, Sweden, and International Centre for Theory of Quantum Technologies, University of Gdańsk, Poland
AKSHATA SHENOY H., International Centre for Theory of Quantum Technologies, University of Gdańsk, Poland
ANTONIO MANDARINO, International Centre for Theory of Quantum Technologies, University of Gdańsk, Poland and Department of Physics Aldo Pontremoli, University of Milan, Italy
A. EGE YILMAZ, Hochschule Luzern, Institut für Finanzdienstleistungen Zug IFZ, Switzerland
THOMAS ANKENBRAND, Hochschule Luzern, Institut für Finanzdienstleistungen Zug IFZ, Switzerland

Machine learning and quantum machine learning (QML) have gained significant importance, as they offer powerful tools for tackling complex computational problems across various domains. This work gives an extensive overview of QML's uses in quantitative finance, an important discipline in the financial industry. We examine the connection between quantum computing and machine learning in financial applications, spanning a range of use cases including fraud detection, underwriting, Value-at-Risk, stock market prediction, portfolio optimization, and option pricing by overviewing the corpus of literature concerning various financial subdomains.

CCS Concepts: • **General and reference → Surveys and overviews**; • **Hardware → Quantum computation**; • **Applied computing → Physics**; **Economics**; • **Computing methodologies → Machine learning**.

Additional Key Words and Phrases: Quantum Machine Learning, Machine Learning, Finance, Quantum Computation

1 INTRODUCTION

In recent years, the convergence of quantum computing (QC) and machine learning (ML) has sparked significant interest across various domains, with finance being no exception. Quantum machine learning (QML) is a branch of ML that harnesses the principles of quantum mechanics to process and analyze data more efficiently. By leveraging the unique properties of quantum systems, such as superposition and entanglement, QML techniques have the potential to improve traditional methodologies in finance. These techniques promise to enhance predictive capabilities in areas such as portfolio optimization, market prediction, trading, pricing, and risk management. As quantum computers continue to evolve and become more accessible, the integration of QML into finance applications is expected.

This review paper delves into this burgeoning field of QML in finance, offering insights into its diverse technologies, applications, and potential implications. The paper is structured into

Authors' Contact Information: Piotr Mironowicz, piotr.mironowicz@gmail.com, Department of Algorithms and Systems Modelling, Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, Narutowicza 11/12, Gdańsk, 80-233, Poland and Department of Physics, Stockholm University, Roslagstullsbacken 21, Stockholm, 114 21, Sweden and International Centre for Theory of Quantum Technologies, University of Gdańsk, Jana Bażyńskiego 1A, Gdańsk, 80-309, Poland; Akshata Shenoy H., International Centre for Theory of Quantum Technologies, University of Gdańsk, Jana Bażyńskiego 1A, Gdańsk, 80-309, Poland; Antonio Mandarino, antonio.mandarino.work@gmail.com, International Centre for Theory of Quantum Technologies, University of Gdańsk, Jana Bażyńskiego 1A, Gdańsk, 80-309, Poland and Department of Physics Aldo Pontremoli, University of Milan, Via Celoria 16, 20133 Milan, Italy; A. Ege Yilmaz, Hochschule Luzern, Institut für Finanzdienstleistungen Zug IFZ, Suurstoffi 1, 6343 Rotkreuz, Switzerland; Thomas Ankenbrand, Hochschule Luzern, Institut für Finanzdienstleistungen Zug IFZ, Suurstoffi 1, 6343 Rotkreuz, Switzerland.

several sections, each focusing on different facets of this intersection. Firstly, in sec. 2, we explore various quantitative finance use cases where QML techniques can be applied, including portfolio optimization, market prediction and trading, pricing, and risk management. Subsequently, we delve into the underlying algorithms and ML methodologies that form the basis for these applications. Following this, the paper delves into the realm of QC and algorithms, providing a foundational understanding of QC principles and covering essential topics such as quantum gates, qubits, quantum circuits, and quantum algorithms.

The crucial part of the review is the comprehensive literature review given in sec. 5, which examines the existing body of research on various aspects of QML in finance. Within this part, we explore specific applications, including portfolio optimization in sec. 5.1, market prediction and trading strategies in sec. 5.2, pricing in sec. 5.3, risk management in sec. 5.4, shedding light on the current state of the art and potential future directions. By providing a holistic overview of the field, this review aims to stimulate further research and facilitate advancements at the intersection of QC, ML, and quantitative finance. Whereas our review concentrates on QML, there exist other reviews concerning QC in finance [32, 175].

2 QUANTITATIVE FINANCE USE CASES

This section provides an overview of some prominent applications in quantitative finance, including portfolio optimization in 2.1, market prediction and trading in 2.2, pricing in 2.3, and risk management in 2.4. Each of these areas plays a vital role in finance, and can at best benefit from the use of quantum computing.

2.1 Portfolio Optimization

The science of building an investment portfolio to maximize returns while lowering risk is known as portfolio optimization. To achieve the best possible balance between risk and reward, a complex process of carefully choosing a wide variety of assets, including stocks, bonds, and other financial instruments, is involved. A key tenet of this strategy is diversification, which distributes assets across a variety of (uncorrelated or low correlated) asset classes and industries, thereby lowering the portfolio's total risk. [105]

To assess the risk-adjusted return of an investment or portfolio, the *Sharpe Ratio* is a commonly used metric. It is computed by dividing the excess return of the investment (or portfolio) relative to the risk-free rate by the standard deviation of the investment's returns. It was created by William F. Sharpe [223]. The formula for the Sharpe Ratio is as follows:

$$r_{\text{Sharpe}} \equiv \frac{R_p - R_f}{\sigma_p}, \tag{1}$$

where R_p is the expected portfolio return, R_f is the risk-free rate, and σ_p is the standard deviation of portfolio returns. A higher Sharpe Ratio indicates a better risk-adjusted return, as it shows how much excess return an investor receives for the extra volatility taken on compared to a risk-free asset. Thus, the Sharpe Ratio plays a crucial role in helping investors make informed decisions about their investments [87, 224].

An all-encompassing approach to portfolio optimization ought to take crucial elements into account such as transaction costs, liquidity, and constraints set by the investor. For example, the ease with which an asset can be purchased or sold without significantly affecting its price is referred to as *liquidity*. Transaction costs comprise a range of expenses related to the purchase and sale of shares, such as taxes and brokerage fees. Furthermore, restrictions might be applied, limiting particular investment kinds or setting particular risk levels based on investor preferences. There are several approaches to achieving an optimal portfolio, each with special characteristics, including risk parity, factor-based models, and mean-variance optimization.

The so-called *mean-variance optimization* was pioneered by Harry Markowitz in the 1950s [162]. This approach entails estimating the expected return and risk (volatility) of each asset based on historical data. Subsequently, the investor constructs what is termed as an *efficient frontier*, encompassing all possible combinations of assets that offer the highest expected return for a given level of risk. To be more precise, this frontier comprises portfolios that meet the unique condition of having no other portfolio with a higher expected return given the same standard deviation of return. The optimal portfolio, in turn, finds its place along this efficient frontier, and its identification relies on determining the combination of assets that delivers the highest return for the investor's specific level of risk tolerance [31]. Implementing the Markowitz mean-variance theory entails estimating the means and covariances of asset returns. Typically, these estimations involve using a sample mean vector and a sample covariance matrix [70].

There are two main types of portfolio optimization problems: unconstrained and constrained [25]. The main difference between unconstrained and constrained portfolio optimization problems is the presence of constraints on certain *weights* parameters in the latter. An unconstrained portfolio optimization problem is a mathematical problem in which the goal is to find the optimal portfolio weights that will maximize or minimize a certain objective function, such as the expected return or the variance of the portfolio, without any constraints on the weights. This means that the weights can take on any value, including negative values, usually meaning a short position on that asset, as long as they result in the optimal portfolio [130]. The constraints in constrained portfolio optimization problem can take many forms, such as limiting the maximum weight of any individual asset, requiring that the weights sum to 1 (to ensure that the portfolio is fully invested), or requiring that the portfolio meet certain regulatory requirements [61]. Constrained portfolio optimization is used more often in practice since it allows for realistic and practical investment scenarios. For example, it may not be realistic to allow for negative weights and to invest more than 100% of the portfolio in a single asset. An example of portfolio optimization of n assets is the following problem:

$$\begin{aligned} &\text{minimize } q \cdot x^\top Q x - \mu^\top x \\ &\text{over } x \in \{0,1\}^n \\ &\text{subject to } 1^\top x = B, \end{aligned} \tag{2}$$

where $q > 0$ reflects the risk appetite of the decision maker, Q is an n by n real matrix specifying covariances between asset returns, μ is a vector with the expected return for each of the assets, and B is the number of assets to be selected that can be interpreted as a budget. The pivotal operation entails computing the inverse of the covariance matrix and solving the associated quadratic optimization problem.

Factor-based models present another popular approach to portfolio construction. These models are utilized by investors to estimate the riskiness and relationships between securities in a portfolio. These models take into account various factors, including value, size, momentum, and quality. By incorporating these factors into the optimization process, investors aim to capture additional explanations of return beyond those explained by traditional asset classes.

An alternate approach to portfolio optimization is provided by the *risk parity* [211], whose main objective is to distribute risk equally among diverse asset classes.

Online portfolio optimization is a dynamic approach to investment management that takes into account new information as it becomes available. This is in contrast to traditional portfolio optimization, which assumes that all relevant information is known at the time the portfolio is created. Online portfolio optimization algorithms can adjust the portfolio weights in real-time, allowing for more efficient and effective risk management [113, 152]. *Portfolio rebalancing* is the process of realigning the weightings of a portfolio of investments. It involves periodically buying or selling assets in the portfolio to maintain a desired asset allocation.

2.2 Market Prediction and Trading

Market prediction refers to the process of forecasting future trends and movements in financial markets. It involves analyzing various factors such as historical data, economic indicators, market sentiment, and other relevant information to make predictions about the direction of prices, asset values, and overall market conditions. Market prediction include fundamental analysis, technical analysis, quantitative modeling, sentiment analysis, and microstructure analysis [79, 181]. *Fundamental analysis* involves examining the underlying factors that influence the value of an asset or market, by considering economic indicators, industry trends, company financials, and other relevant information to assess the intrinsic value of an asset [23, 245]. *Technical analysis* focuses on studying historical price patterns and market trends to predict future price movements with the aid of tools such as statistical models, indicators, and charts, used to identify patterns that suggest potential selling or buying opportunities [178]. *Quantitative modeling* involves using mathematical models and statistical techniques to analyze datasets and identify patterns or relationships between variables sometimes with support of AI methods [222]. *Sentiment analysis* is a technique that aims to gauge market sentiment or investor emotions by analyzing news articles, social media posts, and other sources of information to monitor the overall sentiment, to assess whether investors are optimistic or pessimistic about the market [47, 131]. *Microstructure analysis* delves into the intricate details of how orders are executed, prices are formed, and liquidity is provided in financial markets [140]. By studying the dynamics of order flow, market impact, and price discovery processes, quantitative analysts can gain insights into market behavior and optimize their trading strategies accordingly. Understanding market microstructure is crucial for developing effective trading algorithms that can navigate the complexities of modern financial markets [85].

Algorithmic trading involves the use of computer algorithms to execute trading strategies at a speed and frequency that is often impossible for human traders [43]. This approach relies on quantitative models to identify trading opportunities based on various factors such as price movements, volume, and market data. Algorithms can be designed to execute trades automatically based on predefined criteria, allowing for rapid decision-making and execution in the financial markets [46, 140]. Market prediction is inherently uncertain and subject to various risks and limitations, as financial markets are influenced by a multitude of factors, including economic conditions, geopolitical events, investor behavior, and unforeseen events which can lead to unexpected market movements that may deviate from predictions. To improve the accuracy of the prediction of the stock market, neural networks (NNs) have been employed [1, 12, 33].

2.3 Pricing

Options are financial derivative contracts that give the right, without imposing an obligation, to either purchase, known as a *call option*, or sell, known as a *put option*, an underlying asset at an agreed-upon price called the *strike*, within a specified timeframe known as the *exercise window* [122]. There are several types of options based on their exercise characteristics and settlement terms. The major types include:

(1) *American* options, which can be exercised at any time before expiration, making them more flexible than European options;

(2) *European* options, which can only be exercised at expiration, which simplifies their pricing compared to American options;
(3) *Bermudan* options, which can be exercised at a set (always discretely spaced) number of times, what makes it intermediate between American and European options;
(4) *Asian* options, or average value options, which have a payoff based on the average price of the underlying asset over a specific period rather than just the spot price at expiration;
(5) *Barrier* options, which come into existence or cease to exist when the underlying asset's price reaches a predetermined barrier level [50].

Optimal stopping theory deals with determining the optimal moment to "stop" or take an action to maximize the expected reward [225, 246]. For instance, American options can be seen as a super-martingale hedging problem for the seller and a stochastic optimal halting problem for the buyer [90].

Because the characteristics that define an option are stochastic, it is often difficult to determine its fair value; for this reason, option pricing often requires the use of numerical methods. Because of its adaptability and efficient handling of stochastic parameters, Monte Carlo simulation is one of the most popular techniques [34]. Monte Carlo simulation is used to simulate possible future outcomes by generating random samples and is particularly useful for complex option pricing problems with multiple sources of uncertainty [202]. Nonetheless, for complex options Monte Carlo methods usually require significant computational resources to provide accurate option price estimates [227]. Scientific research fields in option pricing concentrate on refining current models, creating new pricing techniques, investigating alternative market dynamics hypotheses, and advancing options trading risk management strategies. The goal of *stochastic volatility models* [16] is to increase option pricing accuracy [86] and better represent the dynamics of volatility in financial markets. *Jump diffusion models* incorporate sudden jumps in asset prices into the modeling framework to account for extreme market events that impact option prices [64].

One of the most well-known methods for option pricing is the *Black-Scholes model.* By taking into account variables such as the price of the underlying asset, the strike price, the time to expiration, the risk-free rate, and volatility, it offers an estimate of the value of European-style options [30]. The model assumes that the stock price follows a lognormal distribution, and stock returns a normal distribution; there are no transaction costs or taxes; the risk-free interest rate is constant; and the option can only be exercised at expiration; there is no arbitrage nor dividends [123]. Let us denote by $C(S_t, t)$ the call option price at time t and the stock price S_t at t, K is the strike price of the option, T is the time of maturity of the option, r is the risk-free interest rate, and $N(x)$ is the cumulative distribution function of the standard normal distribution. The model introduces the following formulae:

$$C(S_t, t) = S_t N(d_1) - Ke^{-r(T-t)} N(d_2), \tag{3a}$$

$$P(S_t, t) = Ke^{-r(T-t)} N(-d_2) - S_t N(-d_1), \tag{3b}$$

where

$$d_1 = \frac{\ln(S_t/K) + (r + \sigma^2/2)(T-t)}{\sigma\sqrt{T-t}}, \tag{4a}$$

$$d_2 = d_1 - \sigma\sqrt{T-t}. \tag{4b}$$

The former equation (3a) is for a European call option, and the latter (3b) is for a European put option. A more generic numerical method for the valuation of options is the binomial options pricing model. It is a lattice-based model utilizing discretized time to represent the fluctuating price of the underlying financial instrument. This approach is useful in situations when the closed-form Black-Scholes formula is not feasible. It can be applied to both American and Bermudan options since it employs the instrument's description over a period of time rather than a single point. Although it requires more computing power than the Black-Scholes model, it is often more accurate. In 1979, Rendleman and Bartter separately, and Cox, Ross, and Rubinstein (CRR) formalized it [59, 209].

The Heath-Jarrow-Morton (HJM) framework is a mathematical model to price interest rate derivatives [111]. HJM and its variants are described by the following stochastic differential equation:

$$df(t,T) = \alpha(t,T)\,dt + \sigma(t,T)\,dW(t) \tag{5}$$

where T denotes *maturity*, i.e. the time at which the final payment is due on a financial instrument. $df(t,T)$ represents the instantaneous forward interest rate of a *zero-coupon bond*, i.e. a bond that does not make periodic interest payments but profits to the investor from the difference between the purchase price and the face value, with maturity T. The parameters $\alpha(t,T)$ and $\sigma(t,T)$ are describing the *drift*, i.e. systematic tendency of the stochastic process, and the *diffusion* i.e. random fluctuations over time, respectively. $dW(t)$ denotes a Brownian motion under the risk-neutral assumption [132].

2.4 Risk Management

Value-at-Risk (VaR) is a widely used measure in risk management to quantify the potential loss on an asset or portfolio over a specific time horizon and with a certain confidence level. Selecting a *time horizon* (such as one day or one week) and a confidence level α (such as 95%, 99%) are necessary for the calculation of VaR. The possible loss that could be exceeded with the selected confidence level over the given time horizon is then estimated using past data or statistical models [196]. When the probability of suffering a loss smaller than VaR is (at least) p, and the likelihood of suffering a loss larger than VaR is (at most) $1 - p$, then we say we have p VaR. A loss that is greater than the VaR threshold is called a *VaR breach* [119].

Let us consider a portfolio with K assets, where $\mathbb{L} \equiv (L_1, \cdots, L_K)$ with $L_i \in \mathbb{R}_+$ denote possible losses associated with the relevant asset. The total loss is denoted as $\mathcal{L} \equiv \sum_{i\in[K]} L_i$, and its expected value is $\mathbb{E}[\mathcal{L}] = \sum_{i\in[K]} \mathbb{E}[L_i]$. Let $\alpha \in [0,1]$ be the confidence level. VaR is then defined in the following way:

$$\mathrm{VaR}_\alpha[\mathcal{L}] \equiv \inf_{p\in\mathbb{R}_+} \{\mathbb{P}[\mathcal{L} \le p] \ge \alpha\}. \tag{6}$$

Conditional Value at Risk (CVaR), also known as *expected shortfall*, is the anticipated loss for losses exceeding VaR. CVaR is particularly responsive to extreme events occurring in the tail of the loss distribution.

The *economic capital requirement* (ECR) [186] is a critical risk metric representing the capital needed to maintain solvency at a given confidence level. It is defined as the difference between VaR and the expected loss:

$$\mathrm{ECR}_\alpha[\mathcal{L}] \equiv \mathrm{VaR}_\alpha[\mathcal{L}] - \mathbb{E}[\mathcal{L}]. \tag{7}$$

A pertinent and significant issue revolves around estimating the probability of debtors reimbursing their loans, which is a crucial quantitative matter for banks. Financial institutions typically aim to gauge the creditworthiness of debtors by categorizing them into classes known as *credit ratings*. These institutions have the option to develop their credit rating model or rely on credit ratings provided by major rating agencies. Borrowers are commonly divided into two primary categories based on their creditworthiness: investment-grade borrowers with low credit risk and sub-investment-grade borrowers with higher credit risk. When a borrower's rating declines from investment to sub-investment grade, they are referred to as *fallen angels* [49, 151].

Underwriting is a critical process in the insurance industry where an insurer assesses the risk associated with insuring a particular individual or entity and determines the terms and conditions of the insurance policy, by analyzing various factors such as health history, lifestyle choices, occupation, and more to determine the likelihood of a claim being made [19].

The *Local Outlier Factor* (LOF) algorithm is a widely studied unsupervised anomaly detection (AD) method, frequently employed due to its effectiveness. It operates through three key steps: determining the k-distance neighborhood for each data point x; computing the local reachability density of x; and calculating the local outlier factor of x to determine its abnormality [37, 106]. However, it's important to note that the LOF algorithm can become computationally expensive, especially when dealing with large datasets. *Fraud Detection* (FD), a particular case of AD is a critical area of focus to safeguard financial markets, institutions, and investors from fraudulent activities [117, 253]. Utilizing advanced data analytics and ML algorithms is crucial for detecting anomalies and patterns indicative of fraudulent activities in financial transactions [17]. Understanding the behavioral patterns of individuals or entities involved in financial transactions is essential for fraud detection [14]. Network analysis involves examining the relationships and connections between different entities in the financial system to uncover potential fraud schemes [54]. We refer to [7] for a review of ML methods in fraud detection.

3 MACHINE LEARNING

Machine Learning (ML) is an approach in the development of algorithms and statistical models enabling learning by leveraging data without explicit programming [249]. The algorithm is data-driven and hence adaptable to newer inputs improving their performance over a period. This makes them invaluable when the existence of a suitable algorithm is uncertain but abundant data is available. In the financial sector, ML has made significant impact from algorithmic trading to fraud detection, though their influence has worked both ways [39]. The general steps involved in training a ML algorithm (fig.(1)) are described as follows.

- Formulation of the problem to be solved and collection of data.
- Preprocessing the data [262] for managing the missing values, normalization, dimension reduction using feature selection or feature extraction techniques. In the case of supervised learning, the processed dataset is typically split into training and testing sets.
- Suitable model is selected and trained to predict the target outcomes while optimizing its parameters and performance. The accuracy of the model is assessed with suitable metrics. The model is evaluated on unseen data and further tuning in terms of the hyperparameters is carried out.
- The model is implemented in real-world scenarios with regular maintenance to ensure it is up-to-date.

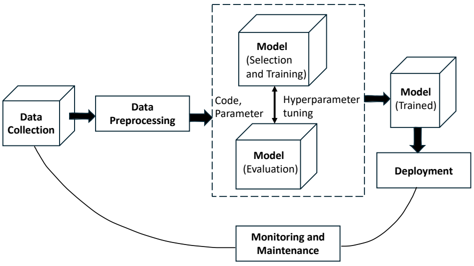

Fig. 1. A general ML process is depicted here.

Generally a dataset is comprised of multiple data points, each characterized by one or more attributes known as features [239]. The total number of features is the dimension of the data. Large dimensional datasets increase the computational complexity associated with training a model known as the “Curse of Dimensionality”. This leads to the risk of overfitting and inefficiency in the training process. As a result, dimension reduction techniques namely feature selection (FS) and feature extraction (FE) emerge as key pre-processes in addressing them [138].

FS reduces the dimensionality of a dataset by identifying and retaining the most relevant features that contribute to the learning process. Data is not merely discarded but methodically selected to expedite training and enhance the interpretability of a model. Effective FS relies on three criteria [143]: *Relevance* is the importance of a feature in making accurate predictions, *Redundancy* eliminates features that do not contribute to additional information, while *Diversity* ensures that the selected data includes features that provide a comprehensive representation of the unprocessed data necessary to address the task. However, if a small subset of data is selected from a very diverse collection of features, the risk of loss of information still persists. FS techniques comprise of: (i) The computationally less intense *Filter methods* which rely on the statistical relation between the feature and the target outcome without any dependency on the ML algorithm. E.g. Correlation coefficient scores, Chi-Square test. (ii) The computationally expensive *Wrapper methods* create a subset of features based on the ML model and achieve the best FS with respect to it. E.g. Forward selection, Backward elimination and Recursive feature selection. (iii) *Embedded methods* integrate FS into the training process of a model itself and perform better than the other two methods. E.g. LASSO and RIDGE regression.

While FS chooses a subset of features from the original data, FE transforms these features into new ones while retaining all the relevant information. Often creating a subspace of lower dimension, the process removes correlations between features[261]. FE methods include: (i) *Principal component analysis* where new variables known as principal components are created as a linear combination of the original feature set. Each principal component is uncorrelated with each other. E.g. Covariance matrix calculation, Eigenvalue and eigenvector computation and sorting. (ii) *Linear discriminant analysis* finds a linear combination of features that maximizes the separation between different classes or groups in the dataset, useful in classification tasks. They include within-class scatter matrix, between-class scatter matrix calculation, eigenvalue and eigenvector computation to name a few. (iii) *t-distributed Stochastic Neighbor Embedding* is a non-linear technique primarily used in exploratory data analysis. It maps high-dimensional datasets to a lower-dimensional space, preserving the similarity relationships between data points. E.g. Probability distribution calculation, similarity calculation and optimization gradient descent.

Finally, generalization of a model, to better capture its predictive powers on unseen data that was not used during the training process but originates from the same distribution [73] is discussed. A generalized model should ideally recognize irregularities and complexities present in this data without overfitting. Patterns and structures identified in the training dataset must be applied accurately on this unseen data. A simpler model is preferred over a complex one when both work equally well with the same predictiveness. Accuracy, precision, recall are certain metrics that analyze the performance of a generalized model. In general, it is known that complexity of the model, quality of the dataset employed in training and regularization techniques affect the generalization process.

ML can be broadly classified into, Supervised learning, Unsupervised learning and Reinforcement learning, based on the algorithms employed for learning. The algorithms are further characterized in terms of their interpretability and explainability to understand why some models outperform others [260]. Interpretability refers to the effect and the cause which gave rise to a particular model and its performance. It determines a model’s predictive capability, in the event a change in its input parameters occur, owing to its implementation in realistic scenarios. Higher the interpretability better is the understanding of how a model arrives at its predictions. This is especially critical for financial transactions where the stakes associated with decisions are greater. For instance, a model's incorrect decision to block a legitimate credit card transaction could have serious consequences in case of an emergency. When a model lacks interpretability, it becomes necessary to employ additional methods to make its decisions comprehendible, leading us to the concept of model explainability. Here, the focus is on elucidating the model's behavior, even if the inner workings driving its predictive powers are not entirely clear. As ML continues to evolve, balancing the performance of models to have a clear understanding of their decision capability is necessary.

3.1 Supervised Learning

Supervised learning (SL) algorithms generate a function that map inputs to desired outputs using labelled datasets. They are divided into different classes of finite order, forming the basis for model training. SL model trains to employ the labelled datasets and associate the correct output to the corresponding input via mathematical analysis [176]. SL can be divided into classification and regression algorithms. A classification model maps its inputs to predetermined discrete outputs. Most common example is classifying emails into ‘spam’ or ‘not spam’ classes. A regression model maps an input to a continuous or a real-valued output. For instance, the value of a house may vary depending on its construction year, area and location. We now briefly discuss some of the SL algorithms.

Linear Regression (LR) predicts outcomes accurately using a set of optimal coefficients. The key assumptions here are that the relationship between the input and output variables is linear (*Linearity*), both the variables are error-free, input variables are independent of each other (*Absence of multi-collinearity*) and normalized (*Normalization*). For correlated inputs, LR tends to overfit while their normalization ensures better predictions. A general LR model can be expressed as, $\mathcal{M}_{\beta_0\beta_1}(X) = \beta_0 + \beta_1 X$, where X is the input variable, β_1 is a d-dimensional vector of coefficients and β_0 is a real number known as the bias coefficient or intercept. This model $\mathcal{M}_{\beta_0\beta_1}(X)$ is utilized to predict the output variables Y ($Y \longleftarrow \mathcal{M}_{\beta_0\beta_1}(X)$). Different coefficients (β_0, β_1) and (β_0', β_1') lead to two different models $\mathcal{M}_{\beta_0\beta_1}(X)$ and $\mathcal{M}_{\beta_0'\beta_1'}(X)$ predicting Y and Y' respectively. Such models are very useful in predicting the financial and economic performances of companies, banks[15], stock market and trading[8]. Though a LR model is simple and easy to interpret, it is not always suitable for classification problems since it predicts continuous-valued outcomes. A more accurate approach would be to employ logistic regression where the output is binary.

Logistic Regression is useful for categorical output variable. An input x is said to belong to a particular class y or not, corresponding to $P(y = 1|x)$ (positive decision) or $P(y = 0|x)$ (negative decision), based on $\sigma(x) = \frac{1}{1+e^{-x}}$ known as the sigmoid function. Here, the relationship between the input and the output is not linear. The choice between linear and logistic regression depends on the nature of the target outcome. Logistic regression is very aptly suited for binary classification in high risk scenario[35] where precise financial decisions are to be made.

Decision Trees are models with tree-like structure for performing decision tasks[177]. They consists of nodes, branches and leaves. The basic node is known as the root and has no incoming edges. Internal nodes have outgoing edges and represent a test on the features. Each branch indicates the outcomes of this test while the leaves are a class or a continuous value. Starting from the root of the tree, data is split into different nodes based on certain criteria using the branches until the final outcome is reached. Decision trees can be categorical or regression type. They are easy to understand and do not require data normalization. Such models very useful in business decision-making tasks[93]. A major disadvantage of such algorithms are risk of overfitting and model instability for small variations in input data.

Support Vector Machines (SVMs) are helpful in linear and nonlinear classification problems[198]. In this formalism, an optimal hyperplane is constructed using support vectors, that clearly separates two classes of data. Support vectors are the closest data points to the hyperplane. A hyperplane is mathematically represented as $\beta_0 + \beta_1.X = 0$ where β_0 is the bias and β_1 is the $p-$dimensional vector representing a set of data points $\{(x_i, y_i)\}_{i=1}^{n}$, n is the length of the sample and $y_n = \pm 1$ depending on the class to which the data point belongs to. For instance, consider the binary classification problem[92], where the aim is to construct a hyperplane that divides the datasets into two classes. All the points corresponding to $y_n = 1(-1)$ belong to the first (second) class and lie on one (other) side of the hyperplane. Although (SVMs) are meant for general linear classification, with the help of Kernel functions, they can be used for non-linear cases as well. Kernel functions map non-linear datasets into a larger dimensional linear space which can then be separated as different classes using a hyperplane. Commonly used kernel functions are polynomial functions, radial basis functions and sigmoid functions. Support vector machines are suitable for small or medium-sized datasets of data mining and have been extensively applied for financial forecast.

Support Vector Regression (SVR) is an extension of SVMs though they are useful for predicting continuous values as opposed to classification tasks. Suppose $\{(x_i, y_i)\}_{i=1}^{n}$ is the available training data set[13], then the basic idea of the SVR is to find a function $f(x)$ that predicts the target values with no more than ϵ deviation from the actual values. A best fit with ϵ-threshold in contrast to a best hyperplane is looked for. Represented as a tubular region around the regression line, the ϵ-region encompasses the errors points which are not penalized. Any other point lying outside this region is penalized. In addition to the ϵ-threshold, the function $f(x)$ is expected to be as flat as possible. The trade-off between the $\epsilon-$deviation and the flatness can be captured by the C parameter. Similar to SVMs, the input dataset can be transformed into a higher dimensional space using common Kernel functions. The most common use cases of this formalism w.r.t. finance includes stock market exchange and currency conversion[114].

Neural Networks (NNs) perform computations using artificial neurons which are several interconnected nodes that mimic a human brain and its functioning. A NN comprises of an input layer for receiving the input data, an output layer for accessing the targeted output and several in between hidden layers for computation and FE depending on the complexity of the problem[55]. Neurons in each layer process inputs received by them and pass their outputs to the next layer for further processing. Associated with each neuron is an activation function (also known as the transfer function) which defines the learning ability of a neuron. This function indicates whether a neuron has to be activated or not for predicting the targeted outcomes. Typical examples of activation functions are the sigmoid and rectified linear unit. The importance of each neuron can be determined by its weight, bias and the activation function. Upon obtaining the final output from the network in a single run, the network re-calculates these parameters for each neuron via a feedback process, to minimize error propagation and predict accurate results. A stable NN is obtained after iterating over several forward and backward propagations. NNs are broadly classified as convolutional NNs, recurrent NNs and deep learning NNs. Though primarily employed for pattern recoginition, speech recognition and natural language processing, these networks can also be used for classification problems similar to logistic regression. For instance, a sigmoid function can be employed to find a separation between classes via hyperplanes (known as the perceptron algorithm). They find application in option trading and financial forecasting.

Finally, w.r.t. classification tasks, we would like to point out some metrics useful in visualizing the performance of any model [89]. The receiver operating characteristic (ROC) curve measures the performance of a classification model for various threshold settings. The curve shows the variation of the true positive rate against the false positive rates at each threshold point. The true positive rate (false positive rate) is defined as the ratio of the true positive (false positive) to the sum of true and false positives. Also known as the sensitivity or Recall, this metric gives the model's ability to infer the proportion of true positives and false positives correctly. An efficient algorithm that computes the points under the ROC curve is known as the Area under the curve (AUC). An average of the performance across all threshold settings can be obtained here. AUC varies from 0 to 1 indicating that the model's predictions are completely incorrect or correct respectively. It is interesting to note that AUC infers the predictive capabilities of a model irrespective of threshold classification. Furthermore, it is noteworthy to mention radial basis functions for SVMs [69] and NNs [171]. These are radially symmetric functions which give the distance of an input from a chosen fixed point. This real-valued function is especially useful for introducing non-linearity in regression and classification models.

3.2 Unsupervised Learning

Unsupervised learning (USL) is applicable in ML for uncategorized datasets where the data inputs do not have corresponding labelled outputs. USL algorithms discover the inherent structure in the dataset by identifying latent variables and patterns without explicit feedback or supervision, in contrast to SL. USL algorithms can be classified into clustering and Association algorithms.

Clustering algorithm is a fundamental USL technique where data points are organized into distinct groups (clusters) based on their similarities and differences [174]. Clustering algorithms are versatile and can be categorized as follows: *Exclusive Clustering* where each data point belongs to exactly one cluster. E.g. K-Means clustering assigns data points to the nearest cluster center. In *Overlapping Clustering* a data point can belong to one or multiple clusters. E.g. Fuzzy C-Means where data points have degrees of belonging to various clusters. *Hierarchical Clustering* builds a hierarchy of clusters, either agglomeratively (bottom-up) or divisively (top-down) continuously merging and reorganizing them during the training process. E.g. Nested structures in datasets. *Probabilistic Clustering* assigns data points to clusters based on their probability of belonging assuming different probability distributions. E.g. Gaussian mixture models.

Association Algorithms find correlations, patterns or relationships between data points within large datasets. A combination of one or more data points that appear in the dataset known as Itemsets, meeting a minimum support threshold, are identified. Support refers to how often a data point occurs in the dataset. Association or relationship rules (If-Then statements) are then built based on how often these itemsets occur. For instance, Market basket analysis [120], where patterns in customer purchases are identified and analyzed is an apt example. The commonly known association algorithms are the A-Priori algorithm employing bottoms-up approach in mining frequent itemsets and the FP-Growth which uses a tree-like structure. The algorithms are important for credit risk assessment where correlations between various financial behaviors and credit risk are identified.

A significant aspect of USL is generative modeling, which focuses on understanding and capturing the internal structure of data. Generative models learn the joint probability distribution of input data, as opposed to discriminative models that focus only on the conditional probability distribution. In the context of finance [170], models like Generative Adversarial Networks (GANs), consisting of a generator and discriminator, generate new and artificial data samples that mimic the real data. This is invaluable in realistic financial scenarios for augmenting datasets where real data is scarce or sensitive. Similarly, Variational Autoencoders are powerful in feature learning and dimensionality reduction. They can reconstruct input data after compressing it into a lower-dimensional subspace, useful for identifying key features in complex financial datasets [257]. Other considerations in USL are anamoly detection, dimensionality reduction and data processing as mentioned before.

3.3 Reinforcement Learning

Consider the context, where a learner is expected to make decisions based on observations to solve a specific problem. Such learners in RL are called *agents*. As the decisions of the agent depend on observations or *states* coming from an *environment*, the observations depend on the decisions or *actions* of the agent. For the vast majority of the problems of practical interest, a priori knowledge of the complete specification of the environment is not available [228], which makes classical optimization methods and supervised learning techniques inapplicable. Instead, this type of information can be revealed by interacting with the environment and developing strategies or *policies* based on the responses from the environment. These responses include the state of the environment and a reward signal based on the actions taken by the agent. The process of taking an action based on an observation and receiving a response as a result is called a *transition* (see Figure 2). RL tasks can be episodic, where transitions lead to a terminal state at a certain point and the task is restarted. They can also be continuous, where this is not the case.

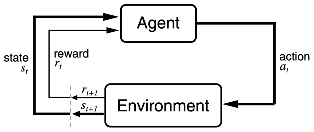

Fig. 2. The transition scheme in RL. An agent acts upon receiving a response from the environment. The response contains a state s_t and a reward r_t at time t. The agent receives a new response as a result of a new action a_t, which consists of a state s_{t+1} and a reward r_{t+1}. This is called a transition. Time steps could refer to arbitrary decision-making iterations, which do not necessarily have fixed intervals of real-time. Picture taken from [228].

Some reinforcement learning algorithms are mentioned in brief here. *Q-Learning* is an approach aimed to develop an optimal action policy for any Markov decision process. It is an off-policy due to random actions taken by the algorithm to gain rewards in a greedy manner. The algorithm uses a Q-function $Q(s, a)$ to estimate the value of an action a taken in a particular state s. This helps the agent in taking further actions to maximize the cumulative reward, sometimes at the cost of sacrificing immediate gains. The Q(s,a) is updated at each step based on the actions taken, rewards assigned and the learning rate[96] known as value iteration. The agent develops an optimal policy by making random actions known as exploitation and observing these actions to increase rewards known as exploration. Q-Learning is simple to implement and works in a model-free environment, though it cannot handle large, complex and continuous states of the environment. In such cases Deep-Q Learning, where Q-Learning is combined with deep learning is preferred[11]. The state-action-reward-state-action algorithm is another important technique which is an on-policy and hence does not make random actions similar to Q-Learning. In contrast to Q-Learning, a policy gradient algorithm learns the policy that maps each state directly to a particular action known as policy iteration. Some noteworthy hybrid algorithms that use both value based and policy based iteration approaches are Soft-Actor-Critic, Twin Delayed Deep Deterministic Policy Gradients and deep deterministic policy gradients[183].

4 QUANTUM COMPUTING AND ALGORITHMS

We will now cover basics of QC and QML. Interested readers seeking a comprehensive introduction to QC can refer to seminal work [71], and recent one [150], and for QML [127].

4.1 Basics of Quantum Computing Methods

Classical computers employ logic gates to execute classical algorithms, analogously a circuit-based quantum computer utilizes quantum gates to run quantum algorithms on quantum bits, known as qubits. The mathematical description of a qubit is a vector in a two-dimensional Hilbert space

$\mathcal{H}) \simeq \mathbb{C}^2$. The Dirac notation is the standard one to denote a qubit, namely a general state can be expressed as:

$$|\psi_1\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{8}$$

in which we have introduced the complex amplitudes α and β, that fulfill the normalization condition $|\alpha|^2 + |\beta|^2 = 1$. The two orthonormal states $|0\rangle$ and $|1\rangle$ form the computational basis for one qubit and conventionally they are the eigenstate of the third Pauli matrix, such that $\sigma_3\,|k\rangle = (-1)^k\,|k\rangle$.

In particular, the identity and the set of the tree Pauli matrices

$$\sigma_0 = \mathbb{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},\ \sigma_1 = \sigma_{\mathrm{x}} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},\ \sigma_2 = \sigma_{\mathrm{y}} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix},\ \sigma_3 = \sigma_{\mathrm{z}} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{9}$$

constitute a basis in which any observable acting on a qubit can be expanded. The linear combination of states in Eq.(8) is known as a superposition of the two basis states, and the squares of the amplitudes account for the probability of detecting the corresponding state. Namely, we have a probability $|\alpha|^2$ ($|\beta|^2$) that the state $|\psi_1\rangle$ collapse in $|0\rangle$ ($|1\rangle$) after a projective measurement.

Similarly, we introduce a register of N qubits

$$|\psi_N\rangle = \sum_{i_1,\ldots,i_N=0,1} \alpha_{i_1,\ldots,i_N}\,|i_1,\ldots,i_N\rangle\,, \tag{10}$$

for which the normalization condition is $\sum_{i_1,\ldots,i_N=0,1} |\alpha_{i_1,\ldots,i_N}|^2 = 1$. Any logic operation on such register is performed via a quantum gate, that is formally expressed as a unitary matrix U such that $U^\dagger U = UU^\dagger = \mathbb{I}$, therefore it belongs to the $\mathbf{SU}(2^N)$ Lie group. Through the set of operators defined in (9), one can define the most relevant class of single and two-qubit parametric operations, respectively:

$$R_\alpha(\theta) = e^{-i\frac{\theta}{2}\sigma_\alpha} \text{ and } C_{\alpha\alpha}(\theta) = e^{-i\frac{\theta}{2}\sigma_\alpha\otimes\sigma_\alpha}, \tag{11}$$

where $\otimes$ denotes the tensor product [160]. They are generally known as rotation and controlled gates and are used to build variational, or parametrized, quantum circuits (PQCs), thanks to the possibility of adjusting the angles θ defining each gate.

Despite significant advancements in designing quantum gates to operate on qubits, a considerable gap still persists between the theoretical results achieved in quantum algorithms, and their practical implementation on quantum processing units. In fact, at the present stage computing systems and simulators remain distant from showcasing any form of quantum advantage in addressing problems emerging outside the purely academic interests and that could impact our daily existence. This gap can be attributed in part to the current era of noisy intermediate-scale quantum (NISQ) technologies [201]. Quantum processing units are constrained by a limited number of qubits and significant levels of noise, including decoherence processes that spoil all the quantum resources requested for the desired speed-up [218, 237]. Techniques to mitigate errors and eventually correct them, such as surface codes and the engineering of logical qubits, have been proposed [41, 97]. Here the main approach is to map a set of physical qubits into a logical qubit, often utilizing specialized network-like circuits for quantum processors composed of logical qubits. However, given the non-unique mapping questions related to the optimal embedding arise and about the scalability of such procedure on different computing architectures.

4.2 Basic building blocks of hybrid algorithms

A *hybrid quantum-classical algorithm* combines elements of both quantum and classical computation to solve computational problems more efficiently than classical methods alone. These algorithms leverage the strengths of quantum computing, such as superposition and entanglement, alongside classical techniques to achieve better performance. A *Quantum Processing Unit* (QPU) is a hardware device designed to perform quantum computations. It consists of qubits, the fundamental units of quantum information, and is used to execute quantum algorithms. The *Quantum Approximate Optimization Algorithm* (QAOA) [84] is a quantum algorithm designed to solve combinatorial optimization problems by preparing a quantum state that encodes the solution to the problem and then measuring it to obtain an approximate solution. The *Variational Quantum Eigensolver* (VQE) [236] is a quantum algorithm used to find the ground state energy of a given Hamiltonian by optimizing the parameters of a parameterized quantum circuit. A *Parameterized Quantum Circuit* (PQC) is a quantum circuit with adjustable parameters that can be optimized to solve specific computational tasks. Finally, a *Quantum Support Vector Machine* (QSVM) [208] is a QML algorithm used for classification tasks, leveraging QC principles to perform efficient classification of data points into different classes.

Quantum annealing [128] (QA) is a computational technique that leverages quantum mechanical principles to solve optimization problems. In QA, a quantum system is initialized in a simple, known state and gradually evolved towards a low-energy state that represents the solution to the optimization problem. *Forward annealing* refers to the process of gradually decreasing the system's temperature or energy levels from an initial high value to a low value, allowing it to explore different configurations and settle into the ground state. *Reverse annealing* [184], on the other hand, involves starting from the ground state and gradually increasing the temperature or energy levels to explore different configurations before reaching the desired state. *Simulated quantum annealing* (SQA) [10], also known as quantum-inspired annealing, mimics the behavior of QA using classical computing resources, often by simulating the behavior of a quantum system undergoing annealing. While not as powerful as true QA, the simulated one can still be useful for solving optimization problems more efficiently than classical methods in certain cases.

The *quantum phase estimation algorithm* is a technique for estimating the phase that corresponds to an eigenvalue of a given unitary operator [141]. *Grover* provided an algorithm for unstructured search in [101]. A quantum algorithm for *minimal search* is shown that locates the index y in a table T of size N such that $T[y]$ is minimum with a probability of at least $1 - 1/2^c$, with a time complexity of $O(c\sqrt{N})$ was given in [76].

The Grover searching technique was expanded upon by *Quantum Amplitude Amplification and Estimation* (QAE), which was first presented in [36]. This approach considers a Boolean function $\chi : X \longrightarrow \{0, 1\}$, where x is refered to as "good" if $\chi(x) = 1$ and "bad" otherwise. Consider a quantum algorithm $\mathcal{A}$, $\mathcal{A}\,|0\rangle = \sum_{x\in X} \alpha_x\,|x\rangle$, and let a represent the probability that a good element is obtained if $\mathcal{A}\,|0\rangle$ is measured. Next, assuming that algorithm $\mathcal{A}$ makes no measurements, amplitude amplification is a procedure that enables a good x to be located. Note that $\mathcal{A}$ in Grover's searching algorithm is limited to generating an equal superposition of all members of X, and requires that there is a known unique x such that $\chi(x) = 1$. Whether the value of a is known in advance or not, QAE still functions. After several applications of $\mathcal{A}$ and its inverse, which is proportional to $1/\sqrt{a}$ even in the worst scenario, a "good" x can be identified if the value of a is known. For a wide range of search problems for which there are effective classical heuristics, the quadratic speedup can also be achieved. It is possible to estimate the value of a by combining concepts from Grover's and Shor's quantum algorithms in the *amplitude estimation* process. Applying QAE to the problem of *approximate counting* allows one to estimate the number of $x \in X$ such that $\chi(x) = 1$.

Several ideas have been put forth on how to construct quantum modular exponentiation, multipliers, and adders utilizing a set of basic quantum gates. Reversible versions of well-known classical implementations were the first circuits that were proposed [24, 241]. Further significant developments include [60, 74, 75, 161, 230, 235, 240, 264], see [187, 215] for an overview.

A quantum algorithm, called Harrow–Hassidim–Lloyd (HHL), for solving systems of linear equations efficiently was demonstrated in [108]. For a given linear equation $Ax = b$ with the unknown x, and A with condition number κ, and given matrix M, the HHL computes the expectation value of $x^{\dagger}Mx$. For small κ it was shown that any classical algorithm requires exponentially more time than HHL. Later, the HHL method was used in the work [254], which efficiently determined the quality of a least-squares fit over an exponentially large data set. In many instances, the algorithm could also efficiently identify a concise function that approximated the data to be fitted and bounded the approximation error. In cases where the input data consisted of pure quantum states, the algorithm was employed to provide an efficient parametric estimation of the quantum state, and therefore, it could be utilized as an alternative to full quantum-state tomography, particularly when a fault-tolerant QPU was available.

In [158] it was demonstrated that multiple copies of a quantum system, characterized by a density matrix ρ, could be employed to construct the unitary transformation $\exp(-i\rho t)$. Consequently, this enabled the performance of quantum principal component analysis on an unknown low-rank density matrix. This approach allowed for the determination in a quantum form of eigenvectors associated with the largest eigenvalues, accomplishing this task in exponentially less time compared to any preexisting algorithm.

A quantum algorithm is presented by the authors of [27] for systems of linear ordinary differential equations with constant coefficients, including possibly inhomogeneous instances. This technique delivers an exponential improvement over previous quantum methods by producing a quantum state proportional to the answer at a predetermined final time and attaining polynomial complexity in the logarithm of the inverse error. They are simulating the evolution according to the propagator using a Taylor series and encoding the simulation into a sparse, well-conditioned linear system by leveraging the HHL approach. Their method provides improved numerical stability without requiring extra hypotheses by avoiding the drawbacks of finite difference techniques. Consequently, they present a quantum algorithm for linear differential equations with a complexity of $poly(\log(1/\epsilon))$, marking a substantial exponential improvement over existing methods [26] where the overall complexity remains $poly(1/\epsilon)$ due to the inherent error introduced by the multistep method.

The Feynman-Kac formula establishes a connection between solutions of certain partial differential equations (PDEs) and Markov processes and provides a way to solve certain types of the former by relating them to the behavior of the latter. An algorithm based on variational quantum imaginary time evolution for solving the Feynman-Kac partial differential equation resulting from a multidimensional system of stochastic differential equations was proposed in [6]. The correspondence between the Feynman-Kac PDEs and the Wick-rotated Schrödinger equation was utilized for this purpose.

Exact inference on Bayesian networks is widely acknowledged as a computationally challenging task, characterized by its #P-hard complexity, thus practitioners usually resort to approximate inference techniques when dealing with such networks. These techniques are employed to draw samples from the distribution concerning query variables, given the provided evidence variable values (e). Through the implementation of a quantum adaptation of rejection sampling, a substantial enhancement in efficiency is achieved. For a Bayesian network containing n variables, each with at most m parents per node, a single unbiased sample is obtained with classical resources in time $O(nmP^{-1}(e))$, whereas quantum technologies allow for a square-root speedup to $O(n2^{m}P^{-1/2}(e))$ time per sample [159].

In the paper [220] an algorithm for QPU-based prediction was presented, centered around a linear regression model with least-squares optimization. The scheme concentrated on the machine-learning task of predicting the output for a new input based on provided data point examples, and was adapted to handle non-sparse data matrices representable through low-rank approximations, and incorporated substantial enhancements were made to reduce its dependency on the condition number. The prediction's outcome could be obtained through a single-qubit measurement or harnessed for further quantum information processing tasks. To this end the quantum principal component analysis of [158], as discussed above, was employed. Another quantum algorithm for fitting a linear regression model to a given data set through the least squares approach returning the optimal parameters in classical form was presented in [247]. The algorithm, once executed, fully determined the fitted model, allowing for cost-effective predictions on new data; it was able to operate on data sets with non-sparse design matrices. Its runtime was characterized by a polynomial dependence on the logarithm size of the data set, the number of adjustable parameters d, the condition number of the design matrix κ, and the desired precision in the output. It was also established that the polynomial dependencies on d and κ were essential, indicating that significant improvements to the algorithm were unattainable. Furthermore, a complementary quantum algorithm was introduced to estimate the quality of the least-squares fit without explicitly computing its parameters.

A Determinantal Point Process (DPP) is a type of stochastic process characterized by a probability distribution, which is expressed as a determinant of a certain function. They occur in quantum physics and random matrix theory, and provide effective algorithms for tasks such as sampling, marginalization, conditioning, and other inference operations, and thus are important for ML [66, 145]. Their quantum versions were discussed in [135], where novel QML algorithms based on quantum subspace states, offering advancements in quantum linear algebra are introduced. The work considers 3 algorithms. The first algorithm facilitates quantum determinant sampling, achieving a significant speedup compared to classical methods. The second algorithm focuses on quantum singular value estimation for compound matrices, potentially yielding exponential improvements in efficiency. Lastly, the third algorithm reduces the circuit depth for quantum topological data analysis, enhancing computational efficiency.

Recently, orthogonal NNs have emerged as a novel NN architecture that enforces orthogonality on the weight matrices. The characteristic of orthogonality in the trained model weights is utilized to prevent redundancy in the acquired features [133, 153]. The paper [148] introduced two related novel quantum methods for NNs aimed at enhancing performance in ML applications. The first method, called quantum orthogonal NN, utilizes a quantum pyramidal circuit to implement orthogonal matrix multiplication, with efficient training algorithms for both classical and quantum hardware. The second method, quantum-assisted NNs, employs QPU for inner product estimation during inference and training of classical NNs. Extensive experiments on medical image classification tasks demonstrate similar accuracy levels between quantum and classical NNs, suggesting the potential usefulness of quantum methods in visual tasks as quantum hardware advances.

Several recent papers have explored the application of QC principles to sentiment analysis in natural language processing. The work [91] proposed a method based on the Lambeq toolkit, while [147] introduced a quantum-inspired fully complex-valued NN, and [248] presented a quantum-like implicit sentiment analysis approach using sememes knowledge. Liu et al. (2023) [157] surveyed various quantum-cognitively inspired sentiment analysis models. Additionally, Sharma et al. (2022) [221] conducted a comparative study between classical and QML models for sentiment analysis. However, before they become effective for finance, the application of quantum sentiment analysis must wait for further advancements.

A model known as the Hybrid classical-quantum Autoencoder was put forth in [216]. It combined the functions of a PQC placed in the bottleneck of a classical autoencoder (AE). Performance was improved in terms of F1 score, recall, and precision once the PQC was included. The benefits of QC in unsupervised AD are illustrated in [106]. Using QAE and minimal search, the k-distance neighborhood of each data point is identified. Using a quantum multiply-adder, the local reachability density of every data point is computed in parallel. Grover's search for anomalous data points and amplitude estimation are used in parallel to determine each data point's local outlier factor. The method achieves polynomial speedup in the number of data points and exponential speedup in the dimension compared to its traditional equivalent. The work [238] presents a semisupervised AD method based on the SVR with quantum kernel reconstruction loss (QSVR). This model is evaluated against a classical autoencoder, a quantum autoencoder, and against an SVR with an RBF kernel. QSVR outperforms all these models, earning the greatest mean AUC over all data sets. The models are thoroughly benchmarked on ten real-world AD data sets and one toy data set. QSVR performs better on 9 out of 11 data sets compared to the quantum autoencoder. QML for FD in Phishing URL was analyzed in [18].

Adiabatic Quantum Algorithms (AQAs) is a type of quantum algorithm that uses adiabatic evolution, a gradual change in the system's Hamiltonian, to find the ground state of a problem. The basic idea is to start with a simple Hamiltonian that can be easily prepared and solved, and then gradually change it to the problem Hamiltonian, which encodes the solution to the problem. The system will remain in the ground state throughout the evolution if the change is slow enough, and the final state will be the solution to the problem [4]. Counterdiabatic (CD) driving is a technique used to speed up AQAs. The concept of CD driving was first introduced in physical chemistry in 2003 [65] and reintroduced in [28]. The key idea behind CD driving is to introduce additional terms in the Hamiltonian of the system that counteract the unwanted transitions. By carefully designing these counterdiabatic terms, it is possible to drive the system along a specific path that enhances the evolution towards the desired final state [22]. CD driving has been applied e.g. in QA [192]. A review of the basic results for quantum error mitigation techniques in hybrid quantum-classical algorithms is given in [83]. The issue of implementing QAOA in near-term devices is discussed in [263].

4.3 Quantum Machine Learning

Quantum machine learning (QML) has emerged as a novel paradigm at the intersection of quantum computing and machine intelligence, aiming to enhance computation speed-up within the current NISQ era using both quantum and classical computational systems and algorithms [29, 219, 255]. Alas, at the present stage the broader field of quantum machine intelligence [68] aiming to develop a learning theory in the quantum domain has proficiently focused on the subfield of QML. In this area, researchers operate at the frontiers of the available quantum hardware and classical ML, even if the first milestones in QML beared in mind the architecture of a fault-tolerant quantum computer and of a Quantum random-access memory [108]. Aiming at a comprehensive review of all the relevant contributions in the field is still a Sisyphean endeavor due to the escalating number of publications and patents. However, it signals a growing scientific and industrial interest. Therefore, in Section 4.3 we briefly review the algorithms that are the most promising for application in quantitative finance, in our opinion.

Several attempts to translate learning models resembling NNs into the quantum domain, with the primary objective to have a cost function of the form

$$\mathcal{L}(\mathbf{x}) = \phi(W\mathbf{x} + b) \tag{12}$$

that could be linked to the unitary transformations that is the core of a quantum circuit and aiming to harness the benefits of quantum information processing.

However, recently the concept of quantum neural networks (QNNs) encompass an entire class of QML algorithms that are based on a combination of parametrized quantum circuits, that are subsequently optimized or trained with the aid of a classical processing unit. In particular, the link with the NN terms used in classical computer science stems from the fact that in the quantum case the hidden layers of the network are composed by the sequential action of a given ensemble of parametric gates. The main building blocks of a QNN are a preparation routine or feature map [217] in which the data are encoded in a quantum state, the second part of the circuit is the proper variational model containing the gates that are eventually optimised to fulfil the required learning task. The procedure follows the quest of minimizing a loss function minimisation, that is a function of specific measurement outcomes performed on the state. In formula, if we denote by $\mathbf{x}$ the data we want to encode in our state then the unitary describing the feature map will be $U_{\mathbf{x}}$, while the parametric circuit described by the unitary $V(\theta)$ will contain set of trainaible weights θ that will be classically optimized. Therefore, the final N-qubit state on which the loss function will be computed reads: $V(\theta)U_{\mathbf{x}}\,|0\rangle^{\otimes N}$.

Despite the fact that the field is rapidly developing, we refer the interest reader to [44], where a detailed description of the state of the art with the major challenges facing the quantum machine learning algorithms are thoroughly discussed.

5 LITERATURE REVIEW

The body of literature concerning quantum computing in finance is vast and multifaceted. Notably, several authors, including [3, 77, 104, 115, 126, 188, 199], have presented thorough summaries of the uses of quantum computing in finance. Our literature review, on the other hand, focuses on the use of QML in the finance domain. While a majority of the sources analyzed entail theoretical explorations employing quantum algorithms for optimization objectives, thus potentially enhancing the speed of various machine learning techniques, only a small subset of these sources present tangible hardware or simulator implementations. Moreover, not all sources contextualize their algorithms within financial applications. Nonetheless, our review encompasses both practical financial implementations and theoretical advancements in quantum machine learning, recognizing the potential applicability of theoretical progress to established finance use cases. Recently, Wang et al. (2022) [251] have proposed a Quantum Finance Software Development Kit (QFSDK) to address the complexities of implementing quantum finance calculations, particularly for educational purposes. This kit, developed in Python, offers a tool for students to gain hands-on experience with quantum finance concepts and applications, thereby enhancing their understanding of this field.

5.1 Portfolio Optimization and Quantum Machine Learning

As discussed in sec. 2.1 portfolio optimization is a method that involves selecting the best combination of investments to achieve a specific goal, such as maximizing returns or minimizing risk.

One of the first QML approaches to portfolio optimization based on quantum annealers was demonstrated in [213]. The authors addressed a multi-period portfolio optimization problem utilizing D-Wave Systems' quantum annealer, offering a formulation of the problem and discussing various integer encoding schemes. Their numerical examples showcased high success rates, with the formulation accommodating transaction costs and bypassing the need for inverting a covariance matrix. Furthermore, they highlighted the challenges posed by discrete multi-period portfolio optimization and provided insights into potential enhancements for future scalability. For the study [81], 63 securities listed on the Abu Dhabi Securities Exchange were considered, utilizing the weekly closing value of each over a period of one year. The covariance matrix and expected values were read into the classical CPU with MATLAB along with the budgets and other parameters. The approach tested whether the adoption of the D-Wave QPU allowed for a meaningful increment in computational performance for solving the Markowitz portfolio. D-Wave's quantum optimizer was used to find the optimal allocation of funds. The D-Wave Solver API (SAPI) embedding routine was used to program the dense Ising model into the unique hardware connectivity graph on the D-Wave processor called Chimera.

The authors of [242] endeavored to address the limitation of the work [213], which focused on portfolios of insufficient size to assess the scalability of the chosen approach with respect to problem size. They generated parametrized samples of portfolio optimization problems based on real financial data statistics. The samples were linked to quadratic binary optimization forms programmable in the analog D-Wave Quantum Annealer 2000Q. The performance was compared with a genetic algorithm approach with the following results. After investigating various options to optimize quantum computation, they discovered that seeding the quantum annealer with a solution candidate found by a greedy local search and then employing a reverse annealing protocol yielded the best results in terms of expected time-to-solution as a function of the number of variables for the most challenging instance set; this approach was called the optimized reverse annealing protocol. The authors found the method to be more than 100 times faster on average than the corresponding forward quantum annealing.

In the work [207] by leveraging quantum access to historical return records, the algorithm determines the optimal risk-return tradeoff curve, facilitating the sampling of the optimal portfolio. If the pertinent data is stored in quantum random access memory, leveraging HHL, quantum walk Hamiltonian simulation methods, and the quantum state exponentiation method [139, 158], the matrix pseudo-inverse and quadratic optimization problem can be resolved, potentially achieving a runtime polynomial in $\log(N)$, instead of the classical complexity requiring polynomial time in N. Consequently, this approach enables the determination of the risk-return curve and the unveiling of the quantum state associated with the optimal portfolio. To be more specific, this method concerns the unconstrained portfolio optimization problem. The calculations were performed on a QPU emulator. A work appeared soon afterwards [136] dealt with the constrained portfolio optimization problem with quantum interior point method for second order cone programs [137] generalized to SDP [134]. The method was evaluated with a QPU emulator with Gaussian noise with a dataset containing historical data about the stocks of the S&P-500 companies from the years 2007-2016.

Next, the work [118] applied to the discrete portfolio optimization problem, a different approach than quantum annealing, namely the gate quantum computing model. They evaluate a portfolio rebalancing use case on an idealized simulator of a gate-model QPU, considering characteristics such as trading in discrete lots, non-linear trading costs, and investment constraints. The authors design a novel problem encoding and hard constraint mixers for the Quantum Alternating Operator Ansatz [107] and compare it to the QAOA. Experimental findings indicate that this application is feasible for NISQ hardware, as it can identify portfolios with adjusted returns within 5% of optimal and optimal risk for a small eight-stock portfolio.

Now let us focus on the recent development of the QA approaches to portfolio optimization. The study [98] focuses on benchmarking on portfolio optimization of the QA controls available in the programmable quantum annealer D-Wave 2000Q, including controls for mapping the logical problem onto hardware and scheduling the annealing process. They explore how the controls influence computational performance and error mechanisms by tuning the quantum dynamics. The study evaluates both forward and reverse annealing methods, identifying control variations that optimize performance e.g. in terms of probability of success. In [197] the portfolio optimization over D-Wave is compared with conventional commercial solvers, demonstrating that the QA approach shows promising performance, coming close to the performance of existing solvers for problems of similar size. In [214] a quantum-inspired integer simulated annealing method for portfolio optimization in the presence of discretized convex and non-convex cost functions is presented, yet not executed on a quantum hardware. The paper [149] compares a workflow combining classical preprocessing with modified QUBO models, evaluated on various annealing platforms, including

D-Wave, using real-world stock data. The outcomes from QA show promise, although they fell short of the performance achieved with simulated annealing and digital annealing. This discrepancy may be attributed to various factors such as inherent noise, lack of error correction, or scaling issues. In work [112] portfolio optimization using digitized-counterdiabatic quantum computing is explored. This concept is applied to discrete mean-variance portfolio optimization, demonstrating improved success probabilities compared to variational quantum algorithms like QAOA and DC-QAOA. The study highlights the potential of digitized-counter-diabatic quantum algorithms for finance applications in the NISQ era.

The series of papers by Cohen et al. [56–58] explores the application of classical and quantum algorithms for portfolio optimization based on the Sharpe ratio, a simplified Chicago Quantum Ratio (CQR), then a new Chicago Quantum Net Score (CQNS), and using U.S. equities. In the first paper [57], they investigate portfolio optimization of 40 stocks using the D-Wave Quantum Annealer, exploring various problem formulations based on risk vs return metrics. The second paper [58] extends this investigation to 60 stocks. Finally, in the third paper [56], the authors analyze 3, 171 U.S. common stocks to create efficient portfolios, incorporating classical solvers and QA techniques. These papers collectively demonstrate the potential of both classical and quantum methods for selecting attractive portfolios in the financial domain.

In [172] a dynamic portfolio optimization with a minimal holding period is proposed and was performed on D-Wave 2000Q. The algorithm efficiently samples near-optimal portfolios at each trading step and post-selects to meet the minimal holding constraint. Results indicate that the method produces investment trajectories much closer to the efficient frontier than typical portfolios and can easily adapt to different risk profiles. The work [190] demonstrates how to obtain the best investment portfolio with a given target risk and implement individual investment bands (i.e. minimum and maximum possible investments) for each asset to impose diversification and avoid corner solutions. The study utilizes D-Wave Hybrid and its *Advantage* QPU to find optimal portfolios for assets from S&P100 and S&P500, showing how practical daily constraints in quantitative finance can be implemented with real data under realistic market conditions. More complex indexes like the Nasdaq Composite can be analyzed with the aid of clustering algorithms. In [173], the problem of dynamic portfolio optimization over a period of time is addressed using classical solvers, D-Wave Hybrid QA, VQEs on IBM-Q, and a quantum-inspired optimizer based on Tensor Networks. The comparison is taken on real data from daily prices over 8 years of 52 assets. Results indicate that D-Wave Hybrid and Tensor Networks can handle the largest systems effectively.

The paper [164] proposes a novel QUBO formulation for portfolio optimization, incorporating both the Sharpe ratio and a diversification term. It modifies the all-or-nothing selection approach of each asset as in [242] by introducing in portfolio construction a linear combination of investments on all assets with arbitrarily large precision. Furthermore, a diversification term is introduced to promote investments across multiple sectors, enhancing portfolio resilience. Results obtained using classical QUBO solvers and the D-Wave Leap hybrid classical-quantum solver demonstrate the effectiveness of the proposed approach in maximizing returns while minimizing sector-specific risks. Optimal outcomes are achieved through D-Wave Leap Hybrid in one of the considered scenarios and classical solver in the other one. The study [189] introduces a system, Q4FuturePOP, designed for portfolio optimization with future asset values executed on D-Wave *Advantage System 6.2*, with 5610 qubits and 40134 couplers spread over a Pegasus topology for QUBO. Unlike traditional approaches using historical data, Q4FuturePOP utilizes future asset predictions for formulating the optimization problem. Through preliminary evaluations it demonstrates promising performance, surpassing expert solutions from financial advisors like Welzia Management in some instances.

Now, we move to other QML implementation approaches. The study [5] compares classical ML, specifically restricted Boltzmann machines (RBMs), with variants of quantum circuit Born machines (QCBMs) implemented on ion-trap QPUs, for a probabilistic version out of the portfolio optimization problem. It utilizes time-series pricing data from asset subsets of the S&P500 stock market index to assess the performance of both models. The quantum models demonstrated superior performance compared to RBMs when considering the same number of parameters. The effectiveness of certain HHL enhancements is empirically demonstrated through the application to small portfolio optimization problems, which were executed end-to-end on the Quantinuum System Model H1-2 trapped-ion QPU, in work [258]. In [155] a quantum version of an existing classical online portfolio optimization algorithm [113], leveraging quantum state preparation, inner product estimation, and multi-sampling techniques, is introduced. The quantum algorithm employs quantum maximum finding, and exhibits a quadratic speedup in time complexity relative to the number of assets in the portfolio n, while the transaction cost remains constant in n, making it particularly suitable for practical applications with a large number of assets. The authors systematically detail the transition from the classical to the quantum algorithm, starting from an extended version of the classical algorithm, incorporating a sampling procedure to render the transaction cost independent of n, and ultimately employing quantum inner product estimation and quantum multi-sampling techniques to devise the quantum online portfolio optimization algorithm.

In [42] a VQE in solving portfolio optimization problems using simulators and real QPUs with over 100 qubits provided by IBM is analyzed. Comparisons are drawn with three classical algorithms by backtesting. Findings indicate that quantum algorithms exhibit competitiveness with classical counterparts, with the advantage of efficiently handling a large number of assets on future larger QPUs. The paper [40] employs VQE to tackle portfolio optimization efficiently and defines optimal hyperparameters for VQE execution on actual IBM QPUs. By converting the problem into QUBO, with constraints integrated into the objective function, the study identifies key hyperparameters like ansatzes and optimization methods. Through experiments on simulators and real QPUs, the research demonstrates a strong correlation between solution quality and quantum hardware size. It concludes that with proper hyperparameters and sufficiently sized quantum hardware, VQE can produce solutions close to exact ones, even without error-mitigation techniques, suggesting a promising avenue for future optimization endeavors.

5.2 Market Prediction and Trading using Quantum Machine Learning

A significant challenge in the financial industry revolves around trading and hedging portfolios of derivatives. Recently, an innovative approach has been introduced to address this issue without relying on frictionless and complete market assumptions [38]. In this approach, trading decisions within hedging strategies are modeled as NNs in a reinforcement learning framework. Expanding on this work, [53] adapts the problem studied by [38] to a quantum-native setup. In this quantum framework, market states are encoded into quantum states, and policies and value functions are represented using Quantum Neural Networks (QNNs). The implementation is carried out on the 20-qubit trapped-ion quantum processor, Quantinuum H1, with training conducted through gradient descent. The effectiveness of this quantum approach is compared against the Black-Scholes delta hedge model, and the results show that the QNN policies outperform the traditional model significantly.

The research conducted in [244] delves into metrics utilized for encoding combinatorial search as a binary quadratic model tailored for feature selection. The primary objective is to enhance the model's overall performance, particularly in terms of generalization, model fit, and accuracy when applied to the regression task of price prediction. Through the utilization of quantum-assisted routines with QUBO using the D-Wave Advantage 1.1 sampler, the authors observed notable enhancements in the quality of predictive model outputs, coupled with a reduction in input dimensionality for the learning algorithm across synthetic and real-world datasets.

In [156] the so-called quantum Elman NN (QENN) was investigated. The Elman NN (ENN) itself, is a type of recurrent NN developed by Jeffrey Elman in 1990 [80] to process sequential data and has been widely used in various fields including time series prediction. The ENN consists of an input layer, a hidden layer, a context layer, and an output layer. The input layer receives the classical input data, which is then processed through the hidden layer, which in the quantum case can be a quantum register. The context layer, which can also be a quantum register, stores the previous state of the hidden layer and provides feedback to the network, allowing it to remember past information and learn temporal dependencies in the data. Finally, the output layer produces the network's output, possibly in quantum form, based on the processed information. The learning rates in [156] are tuned using a quantum version of genetic algorithms. The method was applied for the prediction of closing prices on the Nasdaq, BSE Sensex, HSI, SSE, Russell 2000, and TAIEX stock markets. It was shown that QENN can attain the quality of prediction of ENN with the much smaller size of the hidden and context layers.

The study [191] unveils a novel hybrid deep QNN designed for financial forecasting tasks. Central to the approach is an encoder module that converts partitioned financial time series into a sequence of density matrices, followed by the utilization of a deep quantum network to forecast the density matrix at a future time step. The research demonstrates that the maximum price attained by security at a later time can be extracted from the output density matrix. Through extensive experimentation involving 24 securities, the system showcases remarkable accuracy and efficiency across both regression and extrapolation scenarios.

The paper [63] examined the quantum analogs of classical data preprocessing and forecasting utilizing autoregressive integrated moving average (ARIMA) time series models, employing straightforward quantum operators with minimal quantum gate requirements. The study [82] also explored time series forecasting using quantum technologies. They investigated the effectiveness of PQCs employed as QNN for predicting time series signals through simulated quantum forward propagation. Their findings suggest that QNNs can proficiently model time series data while offering the notable advantage of faster training compared to classical machine learning models when executed on QPUs. Another study [210] focusing on time series forecasting introduces two classical-quantum hybrid architectures employing QNNs and Hybrid-Quantum Neural Networks. These architectures integrate quantum variational circuits with specialized encoding schemes, with optimization executed by a classical computer. The experiments were performed on an QPU emulator. Performance validation across four representative forecasting problems (e.g. USD-to-EUR currency exchange rate) demonstrates competitive performance, despite a comparable number of trainable parameters relative to classical solutions.

The study [234] demonstrates two applications within Itaú Unibanco, Latin America's largest bank. Quantum algorithms for DPP were applied to enhance Random Forest models for churn prediction. These QML algorithms improved precision by nearly 6% compared to baseline models. So-called quantum compound NN architectures which utilize quantum orthogonal NNs, were designed for credit risk assessment. The experiments were conducted on IBM 16-qubits platform *guadelupe*. These NN architectures demonstrated superior accuracy and generalization compared to classical fully-connected NNs while requiring fewer parameters.

In [226] QUBO on D-Wave was employed for feature selection, and Principal Component Analysis (PCA) was used for dimensionality reduction. The task of stock price prediction was transformed into a classification problem, and the QSVM was trained to predict price movements. The performance of QSVM was compared with classical models, and their accuracy was analyzed using datasets formulated with QA and PCA. The study focused on predicting stock prices and binary classification for four companies: Apple, Visa, Johnson and Johnson, and Honeywell, using real-time stock data. Various Quantum Computing techniques were compared with their classical counterparts in terms of prediction model accuracy and F-score. QA showcased superior efficacy in extracting the most pertinent features from financial data compared to PCA. However, QSVM did not demonstrate a notable advantage over classical SVM with the provided datasets.

5.3 Pricing and Quantum Machine Learning

The Heath-Jarrow-Morton (HJM) model, widely employed in finance for valuing interest rate derivatives [111], encounters a notable challenge due to its extensive degrees of freedom when describing the evolution of the yield curve. One potential strategy to tackle this challenge involves the application of principal component analysis for factor selection. The use of quantum Principal Component Analysis (qPCA) can effectively reduce the number of noisy factors as shown in [163], facilitating the determination of fair prices for interest rate derivatives. The estimation of principal components for 2×2 and 3×3 cross-correlation matrices, based on historical data for two and three time-maturing forward rates, is executed using the 5-qubit IBMQX2 quantum processor. The results indicate that the algorithm can provide reasonable approximations for the 2×2 case, although the quantum processor faces limitations related to gate fidelities, connectivity, and the number of qubits. Simultaneously, experimental outcomes with simulators suggest that improved results could be achievable with the availability of a lower-level programming interface. Such an interface would enable the customization of quantum algorithm optimization to align with chip constraints, offering a promising avenue for refinement.

In classical machine learning, Generative Adversarial Networks (GANs) excel at generative modeling, an unsupervised learning approach involving a generator and a discriminator engaged in a competitive training process. The introduction of quantum systems, replacing the generator, discriminator, or both, extends this framework into the domain of quantum computing. An exemplary application is demonstrated in the work [265], where quantum-classical hybrid GANs are employed to learn and transfer approximations of probability distributions from classical data to gate-based QPUs. This is an efficient, approximate data loading scheme that requires significantly fewer gates than existing methods. Specifically, a log-normal distribution is learned that models the spot price of an underlying asset for a European call option. Finally, QAE is used to estimate the expected payoff of the option, given the efficient, approximate data loading by the quantum GANs (qGANs). The training and loading are run on an actual QPU, the IBM Q Boeblingen chip with 20 qubits, with the gradient-based optimization of the qGAN parameters taking place on a classical computer.

The earliest quantum attempts at derivative pricing were presented in two papers by Chen [51] and Chen [52] in 2001. In [51] a quantum adaptation of select areas of arbitrage theory, asset pricing, and optional decomposition within financial markets, were presented, operating on finite-dimensional quantum probability spaces. The work resolved certain paradoxes in classical models and re-deriving option pricing formulas. In [52] a quantum model for binomial markets was proposed, preventing arbitrage opportunities and revisiting option pricing formulas from a quantum perspective. In [212] two formulations for finding optimal arbitrage opportunities as a QUBO solved using D-Wave were presented.

An early quantum algorithm for pricing of financial derivatives was presented in [206]. The relevant probability distributions were prepared in quantum superposition, and the payoff functions were implemented via quantum circuits, with the price of financial derivatives extracted via quantum measurements. QAE on an emulator was applied to achieve a quadratic quantum speedup in the number of steps required to obtain an estimate for the price with high confidence. This study combined QAE and the quantum algorithm for Monte Carlo, with the pricing of financial derivatives, and provided a foundation for further research at the intersection of quantum computing and finance.

The paper [88] introduced a hybrid quantum-classical algorithm, inspired by quantum chemistry, for pricing European and Asian options in the Black-Scholes model. By leveraging the equivalence between the pricing PDE and the Schroedinger equation in imaginary time, the algorithm transforms the Black-Scholes PDE into the Heat Equation and represents the option price as a wave function. This wave function is then solved using a hybrid quantum and classical algorithm, incorporating McLachlan's invariance principle [165] to build a quantum circuit of imaginary time evolution. Despite requiring only a few qubits on an emulator, the shallow quantum circuit accurately represented European and Asian call option prices, indicating a promising potential for applying quantum computing techniques in quantitative finance.

The paper [204] introduced a quantum algorithm for European option pricing in finance, employing a unary representation of the asset value. The algorithm first generates the amplitude distribution corresponding to the asset value at maturity using a low-depth circuit; then it computes the expected return with simple controlled gates; and finally employs the QAE. A comparison of unary and binary option pricing algorithms executed on Qiskit emulator and using error maps indicated that unary representation could offer a significant advantage in practice for near-term devices.

In [227] pricing of various types of options, including vanilla options, multi-asset options, and path-dependent options such as barrier options, was examined using the gate-based IBM Q Tokyo quantum device showing a quadratic speed-up compared to traditional Monte Carlo simulations. Complex features in exotic options, such as path dependency with barriers and averages, were addressed. The results rely on QAE, and its variant without phase estimation [229] was demonstrated to reduce the number of gates required for measuring option prices. An effective error mitigation scheme was employed to reduce errors arising from noisy two-qubit gates.

In [72], a quantum least squares Monte Carlo (LSM) approach is introduced that leverages quantum access to a stochastic process, utilizes quantum circuits for computing optimal stopping times, and employs quantum techniques for Monte Carlo simulations. A nearly quadratic speedup in runtime compared to traditional LSM methods is demonstrated, and examples of its application to American option pricing are given. In [168] LSM was applied to Bermudan option pricing. This method approximates the continuation value, a crucial component of Bermudan option pricing, through Chebyshev interpolation. It utilizes values at interpolation nodes estimated by QAE and demonstrates a quadratic speed-up compared to classical LSM. In [203] the preparation of an initial state representing the option price, followed by its evolution using existing time simulation algorithms in Wick's imaginary time-space were employed for pricing options. Due to its utilization of a hybrid variational algorithm, the proposed method was deemed relevant for NISQ QPUs. The method was numerically verified for European options and has potential extensions to path-dependent options like Asian options.

In [169], the authors focus on derivative pricing through the solution of the Black-Scholes partial differential equation using the finite difference method (FDM), a suitable approach for certain types of derivatives but challenged by the so-called curse of dimensionality. They introduce a quantum algorithm for FDM-based pricing of multi-asset derivatives, demonstrating an exponential speedup in dimensionality compared to classical algorithms. Leveraging quantum differential equations solving algorithms [27], the proposed approach addresses the issue of extracting derivative prices from the output state of the quantum algorithm, outlining the calculation process and estimating its complexity. The work optionally used also QAE for the calculation of the so-called nodal continuation values important for Bermudan option pricing. In the subsequent study [144], the same authors make their algorithm feasible to run on a small QPU but by avoiding certain efficiency bottlenecks of embedding derivative price in the amplitude, they use variational quantum simulation to solve the Black-Scholes equation and compute the derivative price from the inner product between the solution and a probability distribution. They employed a QPU emulator. In [154], a quantum Monte Carlo algorithm was proposed to solve high-dimensional Black-Scholes PDEs with correlation and general continuous and piece-wise affine payoff functions. The approach involves uploading the multivariate log-normal distribution and the rotated form of the payoff function; subsequently, QAE is applied. Error and complexity analyses show that the computational complexity grows only polynomially in the space dimension of the Black-Scholes PDE and the reciprocal of the accuracy level, indicating that the algorithm is not afflicted by the curse of dimensionality. The results were verified on a QPU emulator *qfinance* using Qiskit.

In [45] a strategy to optimize the PQC for pricing a specific type of derivatives, known as Target Accrual Redemption Forward, combining elements of an option and a forward contract, was introduced. This strategy was based on an energy-based method proposed in [195]. It combined pre-trained variational circuits with fault-tolerant quantum computing to reduce resource requirements. The target cost function was determined as the energy of the associated quantum harmonic oscillator problem, Gaussian in its ground state [185]. The circuits required to encode these states for different choices of the register size *n* were pre-trained and applicable to any derivative pricing problem, thereby eliminating the need to include training costs in overall resource estimations. The numerical study demonstrated that the state prepared variably approached the target exponentially fast in terms of the number of gate operations. In the study [129] QAE is the primary source of quantum speedup in a model in which the volatility of the underlying asset price depends on the price and time. The paper explores two variants for the state preparation step of QAE: amplitude encoding, where the probability distribution of the derivative's payoff is encoded into probabilistic amplitudes, and pseudo-random number (PRN) type, where sequences of PRNs simulate asset price evolution akin to classical Monte Carlo simulation.

Other noticeable quantum approaches to option pricing, which we yet not classify as QML include [9, 95, 231, 259]. A so-called quantum first fundamental theorem of asset pricing stating the equivalence between no-arbitrage and the existence of a risk-free density operator under which all assets are martingale was given in [20].

5.4 Risk Management and Quantum Machine Learning

In a pioneering work [256], a quantum algorithm has been presented to analyze risk more efficiently compared to traditional Monte Carlo simulations on classical computers. QAE was utilized to price securities and assess VaR and Conditional VaR on a gate-based QPU. The implementation of this algorithm and the trade-off between convergence rate and circuit depth were demonstrated, indicating a near quadratic speed-up compared to Monte Carlo methods as circuit depths increase gradually. Two toy models were employed to demonstrate the algorithm's efficacy. The first model utilized real hardware, the IBM Q Experience, to price Treasury bills, representing short-term debt obligations, under the risk of potential interest rate increases. The second model simulated the algorithm to showcase how a QPU can evaluate financial risk for a two-asset portfolio comprising government debt with varying maturity dates. Both models confirmed the improved convergence rate over Monte Carlo methods.

In paper [62], a QA algorithm in QUBO form for a dynamic asset allocation problem with an expected shortfall constraint is presented. The algorithm, which is dynamic and allows the risk target to emerge from the market volatility, is formulated in a manner suitable for implementation by a quantum annealer like D-Wave.

The work [78] introduces a quantum algorithm based on QAE for VaR estimation and evaluates it on a Qiskit QPU emulator. The algorithm was designed for the efficient estimation of credit risk, surpassing the capabilities of classical Monte Carlo simulations. Specifically, the algorithm focuses on estimating the ECR. The study implements this algorithm for a realistic loss distribution, providing a comprehensive analysis of its scalability to practical problem sizes. It offers insights into the total number of required qubits, the anticipated circuit depth, and the expected runtime under reasonable assumptions regarding future fault-tolerant quantum hardware. The conclusions highlight a quadratic speedup achieved by the quantum algorithm in estimating economic capital requirements, supported by a simulation that considers realistic problem sizes. The scalability and expected runtime are thoroughly examined, with the suggestion that the results extend to more intricate uncertainty models or alternative objectives, such as conditional value at risk, with minimal additional computational overhead. The development of a quantum circuit for the Gaussian conditional independence model is detailed, with the acknowledgment that diverse credit risk models would necessitate their dedicated quantum circuits.

In [116], variational quantum-classical Wasserstein GANs were presented to solve the problems of sampling efficiency limits and GAN training instabilities. The model maintained the structure of the classical discriminative model but substituted a QNN for the Wasserstein GAN generator to ensure that there was no need to prepare high-dimensional classical data in a quantum circuit. In terms of F1 score, the effectiveness of a dataset of credit card fraud was comparable to the traditional method. The work examined with TensorFlow Quantum emulator how to sample noise, layer depth and width, sampling noise, and the initialization strategy for QNN design parameters affected convergence and performance.

Feature selection stands as a challenging and crucial task within machine learning, involving the identification of a subset of pertinent features in a dataset from the original set. This process differs from feature extraction, which generates new features from the original set, capturing essential information in a lower-dimensional space. Utilizing QNNs, a quantum algorithm for feature selection is introduced in [266]. Specifically, QNNs are trained to generate feature subsets that optimize the performance of a predictive model. While any arbitrary classifier and scoring function could be used, the study opts for logistic regression as the classifier and log-loss as the performance score. The efficacy of the QNN feature selection method is assessed using a publicly available real-world credit risk dataset containing 1000 data points that evaluate a customer's creditworthiness based on 20 attributes. The optimizer for training the QNN parameters is a version of the gradient-free method called simultaneous perturbation stochastic approximation. The feature selection algorithm is implemented on superconducting quantum hardware *ibmq montreal* with 27 qubits, demonstrating results that compete with state-of-the-art classical methods and, in certain experiments, surpass them.

In [243], it was found that QPUs were able to solve certain tasks related to Foreign Exchange reserves management, including risk measurement using the quantum Monte Carlo method and Markowitz-like portfolio optimization employing the HHL algorithm and QAOA. However, due to current hardware limitations, the application of QAOA for a task with only five binary variables and only a few of the demonstrations were successful. The application of the quantum Monte Carlo method to risk measurement generated only partially correct results, and the use of the HHL algorithm in portfolio optimization failed. Running the algorithms discussed above on a QPU simulator and on IBM Lima superconducting QPU confirmed the correctness of the implementations. The work also provided a very concise yet comprehensive overview of general quantum computation with a tutorial.

After discussing the risk assessment methods we move to the topic of FD with QML. Early promising developments for FD with QML were discussed in [67], with conclusion that QPU available at the time of the work were not stable and error tolerant for practical problems. In [232], different single-qubit architectures proposed in [193, 194] were analyzed and a novel implementation was presented using the Qiskit. The former trained a single-qubit based on the concept of data re-uploading, allowing the encoding of mathematical functions in the degrees of freedom of a series of gates applied to a single-qubit state. The authors trained a QNN on real data with Qiskit QPU emulator, and benchmarked it against a NN algorithm, particularly using the Kaggle credit card FD dataset with 284807 transactions, out of which 492 are fraudulent [100]. Different accuracies associated with various layer formulations, the use of an initial data loading layer, and different numbers of layers for a specific problem were demonstrated. In [146], more complex QNN models were used on that dataset. The results established classical benchmarks based on supervised and unsupervised ML methods, with average precision chosen as a robust metric for detecting anomalous data. Quantum kernels of different types were employed for performing AD, and it was observed that the method could challenge equivalent classical protocols as the number of features, equal to the number of qubits for data embedding, increased. Simulations with registers up to 20 qubits showed that quantum kernels with re-uploading demonstrated better average precision, with the advantage increasing with system size. At 20 qubits, the quantum-classical separation of average precision was equal to 15%. In [233] QML was employed for fraud identification in digital transactional payments using the datase [100]. The implementation of the QNN was emulated using Python, and it was observed that the classical Neural Network required about 4.7 times more time and achieved lower accuracy (95.37%) compared to QNN. In [252], a QNN was introduced to learn directly from raw images to train a normality model. It was demonstrated that a quantum-classical hybrid solution , executed on the QPU emulator Rigetti's Forest SDK, can outperform its classical counterpart, even when they have the same number of learnable parameters. Other works concerning AD not directly related to finance include [94, 182].

A hybrid system integrating quantum and classical machine learning algorithms for the detection of phishing attacks within financial transaction networks based on the Ethereum blockchain is proposed by [205]. Data is accessed through the Etherscan block explorer, a tool for the open-source public blockchain platform Ethereum. Phishing account labels are derived from public reports on phishing activities, resulting in a dataset of 3 million nodes. Among these, 1165 nodes (0.039%) are identified as phishing, creating a high class-imbalance scenario in the classification task. QNNs and QSVMs are employed for this purpose, with extensive testing of QNNs involving various parametrization schemes and QSVMs implemented on both annealers and gate-based devices. Optimal configurations for the models are determined through simulators, and the study conducts exhaustive experimentation using these optimized models on IBM's 5- and 27-qubit chips, as well as a D-Wave annealer with 5617 qubits. Surprisingly, the results do not indicate a performance improvement with an increased number of qubits. In the optimization of QNNs, the classical optimizer chosen is the gradient-free algorithm known as constrained optimization by linear approximation. The study reveals that stacking and bagging, techniques that capitalize on the complementary strengths of quantum and classical models, lead to improved results. The findings highlight that gate-based QSVMs consistently yield lower false positives, resulting in higher precision compared to other classical and quantum models. This characteristic is particularly valuable in the context of AD problems.

In [99], an application of a QSVM algorithm, utilizing the IBM Safer Payments and IBM Quantum Computers via the Qiskit software stack and real card payment data, for a classification problem in the financial payment industry was presented. A novel method for searching for the best features was explored using the QSVM's feature map characteristics. The results were compared with classical solutions using fraud-specific key performance indicators: Accuracy, Recall, and False Positive Rate, extracted from analyses based on human expertise (rule decisions), and classical machine learning algorithms. The QSVM provided a complementary exploration of the feature space, leading to improved accuracy of the mixed quantum-classical method for FD, despite the use of a drastically reduced data set to fit the current state of Quantum Hardware. The paper [250] proposed a detection system, which was implemented with SVM supplemented with D-Wave QA.

Twelve machine learning methods were further examined to assess their detection performance, and QSVM was contrasted with them on two datasets: a highly imbalanced bank loan dataset [166] (time series) and a moderately imbalanced Israel credit card transactions [200] (non-time series). With the former dataset, the QSVM was found to perform better than the others in terms of speed and accuracy; however, with the latter dataset, its detection accuracy was comparable to that of the others. It was demonstrated for both datasets that feature selection greatly increased the detection speed while only slightly increasing the accuracy. A comparison of four QML models on Qiskit QPU emulator was done in [124], and QSVM performed the best, with F1 scores of 0.98 for both fraud and non-fraud classes. Promising outcomes were also shown by other quantum models.

In [103], a framework was established for QML fairness verification. Adopting the fairness notion which asserts that any two similar individuals must be treated similarly to ensure unbiased treatment, the study explored how quantum noise could potentially enhance fairness. An algorithm was formulated based on Tensor Networks and implemented using Google's TensorFlow Quantum to ascertain whether a (noisy) QML model adhered to fairness principles. Experimental results, including income prediction and credit scoring based on real-world data, validated the utility and effectiveness of the algorithm, particularly for a class of random (noisy) quantum decision models characterized by 27 qubits (resulting in 227-dimensional state space). Subsequently, in [102], the authors extended their work by defining a formal framework for detecting violations of differential privacy.

The work [167] involved a hybrid, quantum multiple kernel learning (QMKL) approach aimed at enhancing classification quality compared to a single kernel method. Robustness testing of QMKL was conducted across various financially relevant datasets utilizing both fidelity and projected quantum kernel techniques. Application of the QMKL encompassed multiple financially related datasets, including HSBC Digital Payment data. Both fidelity quantum kernel [110] and the more recent projected quantum kernel [121] techniques underwent testing in simulation and practical demonstration on quantum hardware *ibm_auckland*. Hardware implementation was optimized using an error mitigation pipeline comprising randomized compiling to mitigate coherent errors and pulse-efficient transpilation to reduce temporal overhead for cross-resonance gates during two-qubit unitary rotations. This pulse transpilation strategy facilitated scaling of the feature space up to 20 qubits on hardware. In [142], the results from [146] were reassessed and utilized as benchmarks, on a Qiskit emulator, to assess the efficacy of two linear time complexity methods based on data size: randomized measurements for quantum kernel measurement [109] and an ensemble method termed variable subsampling [2]. The dataset from [100] was employed for training purposes. It was revealed that while attainable improvements in average precision and F1 score over the classical kernel were observable, they were not deemed very significant. Models utilizing variable subsampling with the inversion test demonstrated stability, whereas those employing the randomized measurement method exhibited high variance. Variable subsampling notably manifested considerable enhancements in training and testing times, suggesting potential performance elevation opportunities through alternate hyperparameters.

In [151], a QML method for fallen-angels forecasting with a quantum-enhanced classifier based on the QBoost algorithm [179, 180] was proposed. This solution was implemented on a neutral atom QPU with up to 60 qubits using a real-life dataset. The proposed classifier, trained on the QPU, achieved competitive performance with 27.9% precision compared to the state-of-the-art Random Forest benchmark, which achieved 28% precision for the same recall of approximately 83%. However, the proposed approach outperformed its classical counterpart in terms of interpretability, employing only 50 learners compared to 1200 for the Random Forest, while maintaining comparable runtimes. In [125], an approach for detecting financial fraud using Quantum GNNs was proposed.

Classical and quantum GNNs benchmark using a real-world financial fraud detection dataset showed that the latter outperformed the former and achieved an AUC of 0.85.

6 CONCLUSIONS

We have provided a comprehensive overview of the applications of QML in finance. Through an exploration of various use cases including portfolio optimization, market prediction, pricing, and risk management, the potential of QML to improve financial analysis and decision-making has been highlighted. By examining the synergy between quantum computing and machine learning, insights into the future of quantitative finance have been elucidated. Despite the promising advancements and competitive performance demonstrated by QML algorithms, challenges such as scalability, hardware limitations, and algorithmic complexity remain. Addressing these challenges will be crucial for realizing the full potential of QML in finance. Overall, this review underscores the importance of continued research and development in QML for advancing quantitative finance and unlocking new opportunities in the financial industry.

Quantum technologies offer promising applications in portfolio optimization, leveraging quantum computing's potential to efficiently solve complex optimization problems. Techniques such as QA and VQEs have been explored to address portfolio optimization challenges. QA algorithms have been employed to find optimal portfolios by minimizing risk while maximizing returns, while VQE algorithms provide a quantum approach to compute the eigenvalues of portfolio matrices. These quantum approaches aim to enhance the efficiency and accuracy of portfolio optimization, potentially outperforming classical optimization methods as quantum computing hardware continues to advance. We also note that in certain cases quantitative finance inspires quantum methods, e.g. in [21, 48] Conditional Value-at-Risk concept was utilized to enhance the efficiency of general quantum optimization techniques.

ACKNOWLEDGEMENTS

This work was initiated when PM was partially, ASH and AM were fully supported by the Foundation for Polish Science (IRAP project, ICTQT, contract No. 2018/MAB/5, co-financed by EU within Smart Growth Operational Programme). PM also acknowledges support from the Knut and Alice Wallenberg Foundation through the Wallenberg Centre for Quantum Technology (WACQT), and from NCBiR QUANTERA/2/2020 (www.quantera.eu) an ERA-Net cofund in Quantum Technologies under the project eDICT. The work of EY and TA was carried out as part of the IFZ FinTech program with financial support from various industry partners and the Lucerne University of Applied Sciences and Arts.

REFERENCES

[1] Ratnadip Adhikari and R. K. Agrawal. 2014. A combination of artificial neural network and random walk models for financial time series forecasting. *Neural Computing and Applications* 24 (2014), 1441–1449.

[2] Charu C. Aggarwal. 2017. *An introduction to outlier analysis.* Springer Cham, The University of Edinburgh. https://doi.org/10.1007/978-3-319-47578-3

[3] Franco D. Albareti, Thomas Ankenbrand, Denis Bieri, Esther Hänggi, Damian Lötscher, Stefan Stettler, and Marcel Schöngens. 2022. A Structured Survey of Quantum Computing for the Financial Industry. https://doi.org/10.48550/ARXIV.2204.10026

[4] Tameem Albash and Daniel A. Lidar. 2018. Adiabatic quantum computation. *Reviews of Modern Physics* 90, 1 (2018), 015002.

[5] Javier Alcazar, Vicente Leyton-Ortega, and Alejandro Perdomo-Ortiz. 2020. Classical versus quantum models in machine learning: insights from a finance application. *Machine Learning: Science and Technology* 1, 3 (2020), 035003.

[6] Hedayat Alghassi, Amol Deshmukh, Noelle Ibrahim, Nicolas Robles, Stefan Woerner, and Christa Zoufal. 2022. A variational quantum algorithm for the Feynman-Kac formula. *Quantum* 6 (2022), 730.

[7] Abdulalem Ali, Shukor Abd Razak, Siti Hajar Othman, Taiseer Abdalla Elfadil Eisa, Arafat Al-Dhaqm, Maged Nasser, Tusneem Elhassan, Hashim Elshafie, and Abdu Saif. 2022. Financial fraud detection based on Machine Learning: A systematic literature review. *Applied Sciences* 12, 19 (2022), 1–24. https://doi.org/10.3390/app12199637

[8] Erdinc Altay and M. Hakan Satman. 2005. Stock market forecasting: artificial neural network and linear regression comparison in an emerging market. *Journal of Financial Management & Analysis* 18, 2 (2005), 18.

[9] Dong An, Noah Linden, Jin-Peng Liu, Ashley Montanaro, Changpeng Shao, and Jiasu Wang. 2021. Quantum-accelerated multilevel Monte Carlo methods for stochastic differential equations in mathematical finance. *Quantum* 5 (2021), 481.

[10] Bruno Apolloni, C. Carvalho, and Diego De Falco. 1989. Quantum stochastic optimization. *Stochastic Processes and their Applications* 33, 2 (1989), 233–244.

[11] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. 2017. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine* 34, 6 (2017), 26–38.

[12] George S. Atsalakis and Kimon P. Valavanis. 2009. Forecasting stock market short-term trends using a neuro-fuzzy based methodology. *Expert systems with Applications* 36, 7 (2009), 10696–10707.

[13] Mariette Awad and Rahul Khanna. 2015. *Support Vector Regression.* Apress, Berkeley, CA, 67–80. https://doi.org/10.1007/978-1-4302-5990-9_4

[14] Bart Baesens, Veronique Van Vlasselaer, and Wouter Verbeke. 2015. *Fraud analytics using descriptive, predictive, and social network techniques: a guide to data science for fraud detection.* John Wiley & Sons, Online ISBN:9781119146841. https://doi.org/10.1002/9781119146841.ch1

[15] Nor Mazlina Abu Bakar and Izah Mohd Tahir. 2009. Applying multiple linear regression and neural network to predict bank performance. *International Business Research* 2, 4 (2009), 176–183.

[16] Clifford A. Ball and Antonio Roma. 1994. Stochastic volatility option pricing. *Journal of Financial and Quantitative Analysis* 29, 4 (1994), 589–607.

[17] Adrian Bănărescu. 2015. Detecting and preventing fraud with data analytics. *Procedia Economics and Finance* 32 (2015), 1827–1836.

[18] Ishan Shivansh Bangroo and Ravi Kumar. 2023. Zérosdetect: Phishing url Detection with Quantum-Driven Zero-shot Learning. In *2023 9th International Conference on Signal Processing and Communication (ICSC).* IEEE, Noida, India, 498–503. https://doi.org/10.1109/ICSC60394.2023.10440725

[19] Caroline Banton. 2023. Underwriting: Definition and How the Various Types Work. https://www.investopedia.com/terms/u/underwriting.asp Accessed: 12 February 2024.

[20] Jinge Bao and Patrick Rebentrost. 2023. Fundamental theorem for quantum asset pricing. (2023). https://doi.org/10.48550/arXiv.2212.13815.

[21] Panagiotis Kl Barkoutsos, Giacomo Nannicini, Anton Robert, Ivano Tavernelli, and Stefan Woerner. 2020. Improving variational quantum optimization using CVaR. *Quantum* 4 (2020), 256.

[22] Francesco Pio Barone, Oriel Kiss, Michele Grossi, Sofia Vallecorsa, and Antonio Mandarino. 2024. Counterdiabatic optimized driving in quantum phase sensitive models. *New Journal of Physics* 26, 3 (mar 2024), 033031. https://doi.org/10.1088/1367-2630/ad313e

[23] Mark P. Bauman. 1996. A review of fundamental analysis research in accounting. *Journal of Accounting Literature* 15 (1996), 1.

[24] David Beckman, Amalavoyal N. Chari, Srikrishna Devabhaktuni, and John Preskill. 1996. Efficient networks for quantum factoring. *Physical Review A* 54, 2 (1996), 1034.

[25] Patrick Behr, Andre Guettler, and Felix Miebs. 2013. On portfolio optimization: Imposing the right constraints. *Journal of Banking & Finance* 37, 4 (2013), 1232–1242.

[26] Dominic W. Berry. 2014. High-order quantum algorithm for solving linear differential equations. *Journal of Physics A: Mathematical and Theoretical* 47, 10 (2014), 105301.

[27] Dominic W. Berry, Andrew M. Childs, Aaron Ostrander, and Guoming Wang. 2017. Quantum algorithm for linear differential equations with exponentially improved dependence on precision. *Communications in Mathematical Physics* 356 (2017), 1057–1081.

[28] Michael Victor Berry. 2009. Transitionless quantum driving. *Journal of Physics A: Mathematical and Theoretical* 42, 36 (2009), 365303.

[29] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. 2017. Quantum machine learning. *Nature* 549, 7671 (2017), 195–202.

[30] Fischer Black and Myron Scholes. 1973. The pricing of options and corporate liabilities. *Journal of Political Economy* 81, 3 (1973), 637–654.

[31] Taras Bodnar and Wolfgang Schmid. 2009. Econometrical analysis of the sample efficient frontier. *The European journal of finance* 15, 3 (2009), 317–335.

[32] Adam Bouland, Wim van Dam, Hamed Joorati, Iordanis Kerenidis, and Anupam Prakash. 2020. Prospects and challenges of quantum finance. (2020). https://doi.org/10.48550/arXiv.2011.06492.

[33] Melek Acar Boyacioglu and Derya Avci. 2010. An adaptive network-based fuzzy inference system (ANFIS) for the prediction of stock market return: the case of the Istanbul stock exchange. *Expert Systems with Applications* 37, 12 (2010), 7908–7912.

[34] Phelim P. Boyle. 1977. Options: A monte carlo approach. *Journal of Financial Economics* 4, 3 (1977), 323–338.

[35] Philippe Bracke, Anupam Datta, Carsten Jung, and Shayak Sen. 2019. *Machine learning explainability in finance: an application to default risk analysis.* Technical Report 816. Bank of England Working Paper.

[36] Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. 2002. Quantum amplitude amplification and estimation. *Contemp. Math.* 305 (2002), 53–74.

[37] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data* (Dallas, Texas, USA) *(SIGMOD '00)*. Association for Computing Machinery, New York, NY, USA, 93–104. https://doi.org/10.1145/342009.335388

[38] Hans Buehler, Lukas Gonon, Josef Teichmann, and Ben Wood. 2019. Deep hedging. *Quantitative Finance* 19, 8 (2019), 1271–1291.

[39] Alan Bundy. 2017. Preparing for the future of artificial intelligence.

[40] Giuseppe Buonaiuto, Francesco Gargiulo, Giuseppe De Pietro, Massimo Esposito, and Marco Pota. 2023. Best practices for portfolio optimization by quantum computing, experimented on real quantum devices. *Sci Rep* 13 (2023), 19434. https://doi.org/10.1038/s41598-023-45392-w

[41] Zhenyu Cai, Ryan Babbush, Simon C. Benjamin, Suguru Endo, William J. Huggins, Ying Li, Jarrod R. McClean, and Thomas E. O'Brien. 2023. Quantum error mitigation. *Rev. Mod. Phys.* 95 (Dec 2023), 045005. Issue 4. https://doi.org/10.1103/RevModPhys.95.045005

[42] Ginés Carrascal, Paula Hernamperez, Guillermo Botella, and Alberto del Barrio. 2023. Backtesting quantum computing algorithms for portfolio optimization. *IEEE Transactions on Quantum Engineering* 5, 3100220 (2023), 1–20.

[43] Álvaro Cartea, Sebastian Jaimungal, and José Penalva. 2015. *Algorithmic and high-frequency trading.* Cambridge University Press, University of Cambridge, UK.

[44] Marco Cerezo, Guillaume Verdon, Hsin-Yuan Huang, Lukasz Cincio, and Patrick J Coles. 2022. Challenges and opportunities in quantum machine learning. *Nature Computational Science* 2, 9 (2022), 567–576.

[45] Shouvanik Chakrabarti, Rajiv Krishnakumar, Guglielmo Mazzola, Nikitas Stamatopoulos, Stefan Woerner, and William J. Zeng. 2021. A threshold for quantum advantage in derivative pricing. *Quantum* 5 (2021), 463.

[46] Ernie Chan. 2013. *Algorithmic trading: winning strategies and their rationale.* Vol. 625. John Wiley & Sons.

[47] M. S. Checkley, D. Añón Higón, and H. Alles. 2017. The hasty wisdom of the mob: How market sentiment predicts stock market behavior. *Expert Systems with Applications* 77 (2017), 256–263.

[48] Bingren Chen, Hanqing Wu, Haomu Yuan, Lei Wu, and Xin Li. 2023. Quasi-binary encoding based quantum alternating operator ansatz. (2023). preprint arXiv:2304.06915.

[49] James Chen. 2021. Fallen Angel: What it is, How it Works, Risks. https://www.investopedia.com/terms/f/fallenangel.asp Accessed: 19 March 2024.

[50] James Chen. 2022. What Is a Barrier Option? Knock-in vs. Knock-out Options. https://www.investopedia.com/terms/b/barrieroption.asp Accessed: 12 February 2024.

[51] Zeqian Chen. 2001. Quantum finance: The finite dimensional case. (2001). arXiv preprint quant-ph/0112158.

[52] Zeqian Chen. 2001. Quantum theory for the binomial model in finance theory. (2001). arXiv preprint quant-ph/0112156.

[53] El Amine Cherrat, Snehal Raj, Iordanis Kerenidis, Abhishek Shekhar, Ben Wood, Jon Dee, Shouvanik Chakrabarti, Richard Chen, Dylan Herman, Shaohan Hu, Pierre Minssen, Ruslan Shaydulin, Yue Sun, Romina Yalovetzky, and Marco Pistoia. 2023. Quantum Deep Hedging. arXiv:2303.16585 [quant-ph]

[54] Chaochang Chiu, Yungchang Ku, Ting Lie, and Yuchi Chen. 2011. Internet auction fraud detection using social network analysis and classification tree approaches. *International Journal of Electronic Commerce* 15, 3 (2011), 123–147.

[55] Rene Y. Choi, Aaron S. Coyner, Jayashree Kalpathy-Cramer, Michael F. Chiang, and J. Peter Campbell. 2020. Introduction to machine learning, neural networks, and deep learning. *Translational Vision Science & Technology* 9, 2 (2020), 14–14.

[56] Jeffrey Cohen and Clark Alexander. 2020. Picking efficient portfolios from 3,171 US common stocks with new quantum and classical solvers. (2020). preprint arXiv:2011.01308.

[57] Jeffrey Cohen, Alex Khan, and Clark Alexander. 2020. Portfolio optimization of 40 stocks using the dwave quantum annealer. (2020). preprint arXiv:2007.01430.

[58] Jeffrey Cohen, Alex Khan, and Clark Alexander. 2020. Portfolio optimization of 60 stocks using classical and quantum algorithms. (2020). preprint arXiv:2008.08669.

[59] John C. Cox, Stephen A. Ross, and Mark Rubinstein. 1979. Option pricing: A simplified approach. *Journal of Financial Economics* 7, 3 (1979), 229–263.

[60] Steven A. Cuccaro, Thomas G. Draper, Samuel A. Kutin, and David Petrie Moulton. 2004. A new quantum ripple-carry addition circuit. (2004). arXiv preprint quant-ph/0410184.

[61] Jakša Cvitanić and Ioannis Karatzas. 1992. Convex duality in constrained portfolio optimization. *The Annals of Applied Probability* 2, 4 (1992), 767–818.

[62] Samudra Dasgupta and Arnab Banerjee. 2019. Quantum annealing algorithm for expected shortfall based dynamic asset allocation. (2019). preprint arXiv:1909.12904.

[63] Ammar Daskin. 2022. A walk through of time series analysis on quantum computers. (2022). preprint arXiv:2205.00986.

[64] María del Carmen Calvo-Garrido and Carlos Vázquez. 2015. Effects of jump-diffusion models for the house price dynamics in the pricing of fixed-rate mortgages, insurance and coinsurance. *Appl. Math. Comput.* 271 (2015), 730–742.

[65] Mustafa Demirplak and Stuart A. Rice. 2003. Adiabatic population transfer with control fields. *The Journal of Physical Chemistry A* 107, 46 (2003), 9937–9945.

[66] Michał Derezinski and Michael W. Mahoney. 2021. Determinantal point processes in randomized numerical linear algebra. *Notices of the American Mathematical Society* 68, 1 (2021), 34–45.

[67] Alessandra Di Pierro and Massimiliano Incudini. 2021. Quantum machine learning and fraud detection. In *Protocols, Strands, and Logic: Essays Dedicated to Joshua Guttman on the Occasion of his 66.66 th Birthday*. Springer, 139–155.

[68] Alessandra Di Pierro and Luca Viganò. 2022. Quantum Machine Intelligence. In *Mapping the Posthuman*. Routledge, 49–55.

[69] Xiaojian Ding, Jian Liu, Fan Yang, and Jie Cao. 2021. Random radial basis function kernel-based support vector machine. *Journal of the Franklin Institute* 358, 18 (2021), 10121–10140.

[70] David Disatnik and Saggi Katz. 2012. Portfolio optimization using a block structure for the covariance matrix. *Journal of Business Finance & Accounting* 39, 5-6 (2012), 806–843.

[71] David P. DiVincenzo. 1995. Quantum computation. *Science* 270, 5234 (1995), 255–261.

[72] João F. Doriguello, Alessandro Luongo, Jinge Bao, Patrick Rebentrost, and Miklos Santha. 2021. Quantum algorithm for stochastic optimal stopping problems with applications in finance. *Proceedings of TQC 2022, LIPIcs* 232, 2 (2021), 1–24. arXiv:https://arxiv.org/abs/2111.15332

[73] Finale Doshi-Velez and Been Kim. 2018. Considerations for evaluation and generalization in interpretable machine learning. *Explainable and Interpretable Models in Computer Vision and Machine Learning* (2018), 3–17.

[74] Thomas G. Draper. 2000. Addition on a quantum computer. (2000). arXiv preprint quant-ph/0008033.

[75] Thomas G. Draper, Samuel A. Kutin, Eric M. Rains, and Krysta M. Svore. 2006. A logarithmic-depth quantum carry-lookahead adder. *Quantum Info. Comput.* 6, 4 (jul 2006), 351–369.

[76] Christoph Durr and Peter Hoyer. 1996. A quantum algorithm for finding the minimum. (1996). arXiv preprint quant-ph/9607014.

[77] Daniel J. Egger, Claudio Gambella, Jakub Marecek, Scott McFaddin, Martin Mevissen, Rudy Raymond, Andrea Simonetto, Stefan Woerner, and Elena Yndurain. 2020. Quantum computing for finance: State-of-the-art and future prospects. *IEEE Transactions on Quantum Engineering* 1 (2020), 1–24.

[78] Daniel J. Egger, Ricardo García Gutiérrez, Jordi Cahué Mestre, and Stefan Woerner. 2020. Credit risk analysis using quantum computers. *IEEE Trans. Comput.* 70, 12 (2020), 2136–2145.

[79] Graham Elliott and Allan Timmermann. 2013. *Handbook of economic forecasting*. Elsevier.

[80] Jeffrey L. Elman. 1990. Finding structure in time. *Cognitive Science* 14, 2 (1990), 179–211.

[81] Nada Elsokkary, Faisal Shah Khan, Davide La Torre, Travis Humble, and Joel Gottlieb. 2017. Financial portfolio management using d-wave quantum optimizer: The case of abu dhabi securities exchange. In *Conference: 2017 IEEE High-performance Extreme Computing (HPEC) - Waltham, Massachusetts, United States of America (IEEE High Performance Extreme Computing Conference (HPEC))*. 1–4.

[82] Dimitrios Emmanoulopoulos and Sofija Dimoska. 2022. Quantum machine learning in finance: Time series forecasting. (2022). preprint arXiv:2202.00599.

[83] Suguru Endo, Zhenyu Cai, Simon C. Benjamin, and Xiao Yuan. 2021. Hybrid quantum-classical algorithms and quantum error mitigation. *Journal of the Physical Society of Japan* 90, 3 (2021), 032001.

[84] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. 2014. A quantum approximate optimization algorithm. (2014). preprint arXiv:1411.4028.

[85] Fastercapital. 2024. Techniques For Analyzing And Interpreting Order Flow Data. https://fastercapital.com/topics/techniques-for-analyzing-and-interpreting-order-flow-data.html Accessed: 12 February 2024.

[86] Fastercapital. 2024. The Volatility Smile: Delving into Stochastic Volatility Models. https://fastercapital.com/content/The-Volatility-Smile--Delving-into-Stochastic-Volatility-Models.html Accessed: 12 February 2024.

[87] J. Fernando. 2024. Sharpe Ratio: Definition, Formula, and Examples. https://www.investopedia.com/terms/s/sharperatio.asp Accessed: 5 March 2024.

[88] Filipe Fontanela, Antoine Jacquier, and Mugad Oumgari. 2021. A quantum algorithm for linear PDEs arising in finance. *SIAM Journal on Financial Mathematics* 12, 4 (2021), SC98–SC114.

[89] Google for Developers. 2022. *Machine Learning*. https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

[90] Hans Föllmer and Alexander Schied. 2016. *Stochastic finance*. De Gruyter, Berlin, Boston. https://doi.org/10.1515/9783110463453

[91] Srinjoy Ganguly, Sai Nandan Morapakula, and Luis Miguel Pozo Coronado. 2022. Quantum natural language processing based sentiment analysis using lambeq toolkit. In *2022 Second International Conference on Power, Control and Computing Technologies (ICPC2T)*. IEEE, IEEE, Raipur, India, 1–6.

[92] Paulo Gaspar, Jaime Carbonell, and José Luís Oliveira. 2012. On the parameter optimization of Support Vector Machines for binary classification. *Journal of Integrative Bioinformatics* 9, 3 (2012), 33–43.

[93] Adrian Gepp, Kuldeep Kumar, and Sukanto Bhattacharya. 2010. Business failure prediction using decision trees. *Journal of Forecasting* 29, 6 (2010), 536–555.

[94] Kumar J. B. Ghosh and Sumit Ghosh. 2023. Exploring exotic configurations with anomalous features with deep learning: Application of classical and quantum-classical hybrid anomaly detection. *Physical Review B* 108, 16 (2023), 165408.

[95] Javier Gonzalez-Conde, Ángel Rodríguez-Rozas, Enrique Solano, and Mikel Sanz. 2023. Efficient Hamiltonian simulation for solving option price dynamics. *Physical Review Research* 5, 4 (2023), 043220.

[96] Abhijit Gosavi. 2009. Reinforcement learning: A tutorial survey and recent advances. *INFORMS Journal on Computing* 21, 2 (2009), 178–192.

[97] Daniel Gottesman. 2020. Technical perspective: Deciphering errors to reduce the cost of quantum computation. *Commun. ACM* 64, 1 (dec 2020), 105. https://doi.org/10.1145/3434157

[98] Erica Grant, Travis S. Humble, and Benjamin Stump. 2021. Benchmarking quantum annealing controls with portfolio optimization. *Physical Review Applied* 15, 1 (2021), 014012.

[99] Michele Grossi, Noelle Ibrahim, Voica Radescu, Robert Loredo, Kirsten Voigt, Constantin Von Altrock, and Andreas Rudnik. 2022. Mixed quantum–classical method for fraud detection with quantum feature selection. *IEEE Transactions on Quantum Engineering* 3 (2022), 1–12.

[100] Machine Learning Group. 2018. *Kaggle credit card fraud detection dataset*. https://www.kaggle.com/mlg-ulb/creditcardfraud

[101] Lov K. Grover. 1996. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th annual ACM symposium on Theory of computing*. Association for Computing MachineryNew YorkNYUnited States, STOC96: ACM Symposium on Theory of Computing Philadelphia Pennsylvania USA, 212–219. https://doi.org/10.1145/237814.237866

[102] Ji Guan, Wang Fang, Mingyu Huang, and Mingsheng Ying. 2023. Detecting violations of differential privacy for quantum algorithms. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security* (<conf-loc>, <city>Copenhagen</city>, <country>Denmark</country>, </conf-loc>) *(CCS '23)*. Association for Computing Machinery, New York, NY, USA, 2277–2291. https://doi.org/10.1145/3576915.3623108

[103] Ji Guan, Wang Fang, and Mingsheng Ying. 2022. Verifying fairness in Quantum Machine Learning. In *Computer Aided Verification*, Sharon Shoham and Yakir Vizel (Eds.). Springer International Publishing, Cham, 408–429.

[104] Yaswitha Gujju, Atsushi Matsuo, and Rudy Raymond. 2023. Quantum Machine Learning on Near-Term Quantum Devices: Current State of Supervised and Unsupervised Techniques for Real-World Applications. (2023). arXiv:2307.00908.

[105] Abhishek Gunjan and Siddhartha Bhattacharyya. 2023. A brief review of portfolio optimization techniques. *Artificial Intelligence Review* 56, 5 (2023), 3847–3886.

[106] Mingchao Guo, Shijie Pan, Wenmin Li, Fei Gao, Sujuan Qin, XiaoLing Yu, Xuanwen Zhang, and Qiaoyan Wen. 2023. Quantum algorithm for unsupervised anomaly detection. *Physica A: Statistical Mechanics and its Applications* 625 (2023), 129018.

[107] Stuart Hadfield, Zhihui Wang, Bryan O'gorman, Eleanor G. Rieffel, Davide Venturelli, and Rupak Biswas. 2019. From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms* 12, 2 (2019), 34.

[108] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. 2009. Quantum algorithm for linear systems of equations. *Physical Review Letters* 103, 15 (2009), 150502.

[109] Tobias Haug, Chris N. Self, and M. S. Kim. 2023. Quantum machine learning of large datasets using randomized measurements. *Machine Learning: Science and Technology* 4, 1 (2023), 015005. https://doi.org/10.1088/2632-2153/acb0b4 arXiv:https://arxiv.org/abs/2108.01039

[110] Vojtěch Havlíček, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow, and Jay M. Gambetta. 2019. Supervised learning with quantum-enhanced feature spaces. *Nature* 567, 7747 (2019), 209–212.

[111] David Heath, Robert Jarrow, and Andrew Morton. 1992. Bond pricing and the term structure of interest rates: A new methodology for contingent claims valuation. *Econometrica: Journal of the Econometric Society* 60, 1 (1992), 77–105.

[112] Narendra N. Hegade, Pranav Chandarana, Koushik Paul, Xi Chen, Francisco Albarrán-Arriagada, and E. Solano. 2022. Portfolio optimization with digitized counterdiabatic quantum algorithms. *Physical Review Research* 4, 4 (2022), 043204.

[113] David P. Helmbold, Robert E. Schapire, Yoram Singer, and Manfred K. Warmuth. 1998. On-line portfolio selection using multiplicative updates. *Mathematical Finance* 8, 4 (1998), 325–347.

[114] Bruno Miranda Henrique, Vinicius Amorim Sobreiro, and Herbert Kimura. 2018. Stock price prediction using support vector regression on daily and up to the minute prices. *The Journal of finance and data science* 4, 3 (2018), 183–201.

[115] Dylan Herman, Cody Googin, Xiaoyuan Liu, Yue Sun, Alexey Galda, Ilya Safro, Marco Pistoia, and Yuri Alexeev. 2023. Quantum computing for finance. *Nature Reviews Physics* 5 (2023), 450–465.

[116] Daniel Herr, Benjamin Obert, and Matthias Rosenkranz. 2021. Anomaly detection with variational quantum generative adversarial networks. *Quantum Science and Technology* 6, 4 (2021), 045004.

[117] Waleed Hilal, S. Andrew Gadsden, and John Yawney. 2022. Financial fraud: a review of anomaly detection techniques and recent advances. *Expert Systems with Applications* 193 (2022), 116429.

[118] Mark Hodson, Brendan Ruck, Hugh Ong, David Garvin, and Stefan Dulman. 2019. Portfolio rebalancing experiments using the quantum alternating operator ansatz. (2019). preprint arXiv:1911.05296.

[119] Glyn A. Holton. 2013. *Value-at-Risk: Theory and Practice* (ii ed.). Academic Press. https://www.value-at-risk.net

[120] Harald Hruschka. 2021. Comparing unsupervised probabilistic machine learning methods for market basket analysis. *Review of Managerial Science* 15, 2 (2021), 497–527.

[121] Hsin-Yuan Huang, Michael Broughton, Masoud Mohseni, Ryan Babbush, Sergio Boixo, Hartmut Neven, and Jarrod R. McClean. 2021. Power of data in quantum machine learning. *Nature Communications* 12, 1 (2021), 2631.

[122] John C. Hull and Sankarshan Basu. 2016. *Options, futures, and other derivatives.* Pearson education india.

[123] CFI Education Inc. 2015. *Black-Scholes-Merton Model.*

[124] Nouhaila Innan, Muhammad Al-Zafar Khan, and Mohamed Bennai. 2024. Financial fraud detection: A comparative study of quantum machine learning models. *International Journal of Quantum Information* 22, 02 (2024), 2350044. https://doi.org/10.1142/S0219749923500442 arXiv:https://arxiv.org/abs/2308.05237

[125] Nouhaila Innan, Abhishek Sawaika, Ashim Dhor, Siddhant Dutta, Sairupa Thota, Husayn Gokal, Nandan Patel, Muhammad Al-Zafar Khan, Ioannis Theodonis, and Mohamed Bennai. 2024. Financial fraud detection using quantum graph neural networks. *Quantum Machine Intelligence* 6, 1 (2024), 1–18.

[126] Antoine Jacquier, Oleksiy Kondratyev, Alexander Lipton, and Marcos Lopez de Prado. 2022. *Quantum Machine Learning and Optimisation in Finance: On the Road to Quantum Advantage.* Packt Publishing Ltd.

[127] Aishwarya Jhanwar and Manisha J. Nene. 2021. Machine Learning: A Quantum Perspective. In *Recent Trends in Intensive Computing.* IOS Press, 342–347.

[128] Mark W. Johnson, Mohammad H. S. Amin, Suzanne Gildert, Trevor Lanting, Firas Hamze, Neil Dickson, Richard Harris, Andrew J. Berkley, Jan Johansson, Paul Bunyk, et al. 2011. Quantum annealing with manufactured spins. *Nature* 473, 7346 (2011), 194–198.

[129] Kazuya Kaneko, Koichi Miyamoto, Naoyuki Takeda, and Kazuyoshi Yoshino. 2022. Quantum pricing with a smile: implementation of local volatility model on quantum computer. *EPJ Quantum Technology* 9, 1 (2022), 1–32.

[130] Ioannis Karatzas, John P. Lehoczky, and Steven E. Shreve. 1987. Optimal portfolio and consumption decisions for a "small investor" on a finite horizon. *SIAM Journal on Control and Optimization* 25, 6 (1987), 1557–1586.

[131] Colm Kearney and Sha Liu. 2014. Textual sentiment in finance: A survey of methods and models. *International Review of Financial Analysis* 33 (2014), 171–185.

[132] Will Kenton. 2023. Heath-Jarrow-Morton (HJM) Model: What it Means, How it Works. https://www.investopedia.com/terms/h/hjmmodel.asp Accessed: 5 March 2024.

[133] Iordanis Kerenidis, Jonas Landman, and Natansh Mathur. 2022. Classical and Quantum Algorithms for Orthogonal Neural Networks. arXiv:2106.07198 [quant-ph]

[134] Iordanis Kerenidis and Anupam Prakash. 2020. A quantum interior point method for LPs and SDPs. *ACM Transactions on Quantum Computing* 1, 1 (2020), 1–32.

[135] Iordanis Kerenidis and Anupam Prakash. 2022. Quantum machine learning with subspace states. *preprint arXiv:2202.00054* (2022).

[136] Iordanis Kerenidis, Anupam Prakash, and Dániel Szilágyi. 2019. Quantum algorithms for portfolio optimization. In *Proceedings of the 1st ACM Conference on Advances in Financial Technologies.* 147–155. arXiv:https://arxiv.org/abs/1908.08040

[137] Iordanis Kerenidis, Anupam Prakash, and Dániel Szilágyi. 2021. Quantum algorithms for second-order cone programming and support vector machines. *Quantum* 5 (2021), 427.

[138] Samina Khalid, Tehmina Khalil, and Shamila Nasreen. 2014. A survey of feature selection and feature extraction techniques in machine learning. In *2014 Science and Information Conference*. IEEE, 372–378.

[139] Shelby Kimmel, Cedric Yen-Yu Lin, Guang Hao Low, Maris Ozols, and Theodore J. Yoder. 2017. Hamiltonian simulation with optimal sample complexity. *npj Quantum Information* 3, 1 (2017), 13.

[140] Robert Kissell. 2020. *Algorithmic trading methods: Applications using advanced statistics, optimization, and machine learning techniques*. Academic Press.

[141] A. Yu Kitaev. 1995. Quantum measurements and the Abelian stabilizer problem. *arXiv preprint quant-ph/9511026* (1995).

[142] Michael Kölle, Afrae Ahouzi, Pascal Debus, Robert Müller, Danielle Schuman, and Claudia Linnhoff-Popien. 2023. Towards efficient quantum anomaly detection: One-class SVMs using variable subsampling and randomized measurements. *preprint arXiv:2312.09174* (2023).

[143] Sotiris Kotsiantis. 2011. Feature selection for machine learning classification problems: a recent overview. *Artificial Intelligence Review* 42, 1 (2011), 157–176.

[144] Kenji Kubo, Koichi Miyamoto, Kosuke Mitarai, and Keisuke Fujii. 2023. Pricing multi-asset derivatives by Variational Quantum Algorithms. *IEEE Transactions on Quantum Engineering* (2023).

[145] Alex Kulesza, Ben Taskar, et al. 2012. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning* 5, 2–3 (2012), 123–286.

[146] Oleksandr Kyriienko and Einar B. Magnusson. 2022. Unsupervised quantum machine learning for fraud detection. *preprint arXiv:2208.01203* (2022).

[147] Wei Lai, Jinjing Shi, and Yan Chang. 2023. Quantum-Inspired Fully Complex-Valued Neutral Network for Sentiment Analysis. *Axioms* 12, 3 (2023), 308.

[148] Jonas Landman, Natansh Mathur, Yun Yvonna Li, Martin Strahm, Skander Kazdaghli, Anupam Prakash, and Iordanis Kerenidis. 2022. Quantum methods for Neural Networks and application to Medical Image Classification. *Quantum* 6 (2022), 881.

[149] Jonas Lang, Sebastian Zielinski, and Sebastian Feld. 2022. Strategic portfolio optimization using simulated, digital, and quantum annealing. *Applied Sciences* 12, 23 (2022), 12288.

[150] Jonathan Wei Zhong Lau, Kian Hwee Lim, Harshank Shrotriya, and Leong Chuan Kwek. 2022. NISQ computing: where are we and where do we go? *AAPPS Bulletin* 32, 1 (2022), 27.

[151] Lucas Leclerc, Luis Ortiz-Gutiérrez, Sebastián Grijalva, Boris Albrecht, Julia R. K. Cline, Vincent E. Elfving, Adrien Signoles, Loïc Henriet, Gianni Del Bimbo, Usman Ayub Sheikh, et al. 2023. Financial risk management on a neutral atom quantum processor. *Physical Review Research* 5, 4 (2023), 043117.

[152] Bin Li and Steven C. H. Hoi. 2014. Online portfolio selection: A survey. *ACM Computing Surveys (CSUR)* 46, 3 (2014), 1–36.

[153] Shuai Li, Kui Jia, Yuxin Wen, Tongliang Liu, and Dacheng Tao. 2019. Orthogonal deep neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, 4 (2019), 1352–1368.

[154] Yongming Li and Ariel Neufeld. 2023. Quantum Monte Carlo algorithm for solving Black-Scholes PDEs for high-dimensional option pricing in finance and its proof of overcoming the curse of dimensionality. *preprint arXiv:2301.09241* (2023).

[155] Debbie Lim and Patrick Rebentrost. 2022. A quantum online Portfolio Optimization algorithm. *preprint arXiv:2208.14749* (2022).

[156] Ge Liu and Wenping Ma. 2022. A quantum artificial neural network for stock closing price prediction. *Information Sciences* 598 (2022), 75–85. https://doi.org/10.1016/j.ins.2022.03.064

[157] Yaochen Liu, Qiuchi Li, Benyou Wang, Yazhou Zhang, and Dawei Song. 2023. A Survey of Quantum-Cognitively Inspired Sentiment Analysis Models. *Comput. Surveys* (2023).

[158] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. 2014. Quantum principal component analysis. *Nature Physics* 10, 9 (2014), 631–633.

[159] Guang Hao Low, Theodore J. Yoder, and Isaac L. Chuang. 2014. Quantum inference on Bayesian networks. *Physical Review A* 89, 6 (2014), 062315.

[160] Antonio Mandarino, Tomasz Linowski, and Karol Życzkowski. 2018. Bipartite unitary gates and billiard dynamics in the Weyl chamber. *Physical Review A* 98 (Jul 2018), 012335. Issue 1. https://doi.org/10.1103/PhysRevA.98.012335

[161] Igor L. Markov and Mehdi Saeedi. 2012. Constant-optimized quantum circuits for modular multiplication and exponentiation. *Quantum Info. Comput.* 12, 5–6 (may 2012), 361–394.

[162] Harry M. Markowits. 1952. Portfolio selection. *Journal of Finance* 7, 1 (1952), 71–91. https://doi.org/10.1111/j.1540-6261.1952.tb01525.x

[163] Ana Martin, Bruno Candelas, Ángel Rodríguez-Rozas, José D. Martín-Guerrero, Xi Chen, Lucas Lamata, Román Orús, Enrique Solano, and Mikel Sanz. 2021. Toward pricing financial derivatives with an ibm quantum computer. *Physical Review Research* 3, 1 (2021), 013167.

[164] Mirko Mattesi, Luca Asproni, Christian Mattia, Simone Tufano, Giacomo Ranieri, Davide Caputo, and Davide Corbelletto. 2023. Diversifying investments and maximizing Sharpe ratio: A novel QUBO formulation. *preprint arXiv:2302.12291* (2023).

[165] A. D. McLachlan. 1964. A variational solution of the time-dependent Schrodinger equation. *Molecular Physics* 8, 1 (1964), 39–44.

[166] Mishra5001. 2019. Credit Card Fraud Detection: Explore All Possibil-ities While Sanctioning a Loan to Any Customer. Kaggle. https://www.kaggle.com/mishra5001/credit-card

[167] Shungo Miyabe, Brian Quanz, Noriaki Shimada, Abhijit Mitra, Takahiro Yamamoto, Vladimir Rastunkov, Dimitris Alevras, Mekena Metcalf, Daniel J. M. King, Mohammad Mamouei, et al. 2023. Quantum multiple kernel learning in financial classification tasks. *preprint arXiv:2312.00260* (2023).

[168] Koichi Miyamoto. 2022. Bermudan option pricing by quantum amplitude estimation and Chebyshev interpolation. *EPJ Quantum Technology* 9, 1 (2022), 1–27.

[169] Koichi Miyamoto and Kenji Kubo. 2021. Pricing multi-asset derivatives by finite-difference method on a quantum computer. *IEEE Transactions on Quantum Engineering* 3 (2021), 1–25.

[170] D. K. Mohanty, Ajaya Kumar Parida, and Shelly Suman Khuntia. 2021. Financial market prediction under deep learning framework using auto encoder and kernel extreme learning machine. *Applied Soft Computing* 99 (2021), 106898.

[171] Gholam Ali Montazer, Davar Giveki, Maryam Karami, and Homayon Rastegar. 2018. Radial basis function neural networks: A review. *Comput. Rev. J* 1, 1 (2018), 52–74.

[172] Samuel Mugel, Mario Abad, Miguel Bermejo, Javier Sánchez, Enrique Lizaso, and Román Orús. 2021. Hybrid quantum investment optimization with minimal holding period. *Scientific Reports* 11, 1 (2021), 19587.

[173] Samuel Mugel, Carlos Kuchkovsky, Escolastico Sanchez, Samuel Fernandez-Lorenzo, Jorge Luis-Hita, Enrique Lizaso, and Roman Orus. 2022. Dynamic portfolio optimization with real datasets using quantum processors and quantum-inspired tensor networks. *Physical Review Research* 4, 1 (2022), 013006.

[174] Samreen Naeem, Aqib Ali, Sania Anam, and Muhammad Munawar Ahmed. 2023. An unsupervised Machine Learning algorithms: Comprehensive review. *Int. J. Comput. Digit. Syst* (2023).

[175] Abha Naik, Esra Yeniaras, Gerhard Hellstern, Grishma Prasad, and Sanjay Kumar Lalta Prasad Vishwakarma. 2023. From portfolio optimization to quantum blockchain and security: A systematic review of quantum computing in finance. *preprint arXiv:2307.01155* (2023).

[176] Vladimir Nasteski. 2017. An overview of the supervised machine learning methods. *Horizons. b* 4 (2017), 51–62.

[177] Arundhati Navada, Aamir Nizam Ansari, Siddharth Patil, and Balwant A. Sonkamble. 2011. Overview of use of decision tree algorithms in machine learning. In *2011 IEEE control and system graduate research colloquium*. IEEE, 37–42.

[178] Rodolfo Toribio Farias Nazário, Jéssica Lima e Silva, Vinicius Amorim Sobreiro, and Herbert Kimura. 2017. A literature review of technical analysis on stock markets. *The Quarterly Review of Economics and Finance* 66 (2017), 115–126.

[179] Hartmut Neven, Vasil S. Denchev, Geordie Rose, and William G. Macready. 2008. Training a binary classifier with the quantum adiabatic algorithm. *preprint arXiv:0811.0416* (2008).

[180] Hartmut Neven, Vasil S. Denchev, Geordie Rose, and William G. Macready. 2009. Training a large scale classifier with the quantum adiabatic algorithm. *preprint arXiv:0912.0779* (2009).

[181] Isaac Kofi Nti, Adebayo Felix Adekoya, and Benjamin Asubam Weyori. 2020. A systematic review of fundamental and technical analysis of stock market predictions. *Artificial Intelligence Review* 53, 4 (2020), 3007–3057.

[182] Hyeondo Oh and Daniel K. Park. 2023. Quantum support vector data description for anomaly detection. *preprint arXiv:2310.06375* (2023).

[183] Junhyuk Oh, Matteo Hessel, Wojciech M Czarnecki, Zhongwen Xu, Hado P van Hasselt, Satinder Singh, and David Silver. 2020. Discovering reinforcement learning algorithms. *Advances in Neural Information Processing Systems* 33 (2020), 1060–1070.

[184] Masaki Ohkuwa, Hidetoshi Nishimori, and Daniel A. Lidar. 2018. Reverse annealing for the fully connected p-spin model. *Physical Review A* 98, 2 (2018), 022314.

[185] Pauline J. Ollitrault, Guglielmo Mazzola, and Ivano Tavernelli. 2020. Nonadiabatic molecular quantum dynamics with quantum computers. *Physical Review Letters* 125, 26 (2020), 260511.

[186] Basel Committee on Banking Supervision. 2010. Basel III: A globalregulatory framework for more resilient banks and banking systems. *Bank for International Settlements, June* (2010). https://www.bis.org/publ/bcbs189.pdf

[187] Francisco Orts, Gloria Ortega, Elías F. Combarro, and Ester M. Garzón. 2020. A review on reversible quantum adders. *Journal of Network and Computer Applications* 170 (2020), 102810.

[188] Román Orús, Samuel Mugel, and Enrique Lizaso. 2019. Quantum computing for finance: Overview and prospects. *Reviews in Physics* 4 (2019), 100028.

[189] Eneko Osaba, Guillaume Gelabert, Esther Villar-Rodriguez, Antón Asla, and Izaskun Oregi. 2023. A quantum computing-based system for Portfolio Optimization using future asset values and automatic reduction of the investment universe. *preprint arXiv:2309.12627* (2023).

[190] Samuel Palmer, Serkan Sahin, Rodrigo Hernandez, Samuel Mugel, and Roman Orus. 2021. Quantum portfolio optimization with investment bands and target volatility. *preprint arXiv:2106.06735* (2021).

[191] Eric Paquet and Farzan Soleymani. 2022. QuantumLeap: Hybrid quantum neural network for financial predictions. *Expert Systems with Applications* 195 (2022), 116583.

[192] Gianluca Passarelli, Vittorio Cataudella, Rosario Fazio, and Procolo Lucignano. 2020. Counterdiabatic driving in the quantum annealing of the p-spin model: A variational approach. *Physical Review Research* 2, 1 (2020), 013283.

[193] Adrián Pérez-Salinas, Alba Cervera-Lierta, Elies Gil-Fuster, and José I. Latorre. 2020. Data re-uploading for a universal quantum classifier. *Quantum* 4 (2020), 226.

[194] Adrián Pérez-Salinas, David López-Núñez, Artur García-Sáez, Pol Forn-Díaz, and José I. Latorre. 2021. One qubit as a universal approximant. *Physical Review A* 104, 1 (2021), 012405.

[195] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O'brien. 2014. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications* 5, 1 (2014), 4213.

[196] Jorion Philippe. 2001. Value at Risk: The new benchmark for managing financial risk. *NY: McGraw-Hill Professional* (2001).

[197] Frank Phillipson and Harshil Singh Bhatia. 2021. Portfolio optimisation using the D-Wave quantum annealer. In *International Conference on Computational Science*. Springer, 45–59.

[198] Derek A. Pisner and David M. Schnyer. 2020. Support vector machine. In *Machine Learning*. Elsevier, 101–121.

[199] Marco Pistoia, Syed Farhan Ahmad, Akshay Ajagekar, Alexander Buts, Shouvanik Chakrabarti, Dylan Herman, Shaohan Hu, Andrew Jena, Pierre Minssen, Pradeep Niroula, et al. 2021. Quantum Machine Learning for finance ICCAD special session paper. In *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*. IEEE, 1–9.

[200] R. Polantizer. [n. d.]. Fraud Detection in Python; Predict FraudulentCredit Card Transactions. https://medium.com/@polanitzer/fraud-detection-in-python-predict-fraudulent-credit-card-transactions-73992335dd90

[201] John Preskill. 2018. Quantum Computing in the NISQ era and beyond. *Quantum* 2 (Aug. 2018), 79. https://doi.org/10.22331/q-2018-08-06-79

[202] Jia Qiuyi. 2009. *Pricing American options using Monte Carlo methods.* Technical Report. Department of Mathematics Uppsala University. arXiv:https://www.diva-portal.org/smash/get/diva2:301061/FULLTEXT01.pdf Accessed: 12 February 2024.

[203] Santosh Kumar Radha. 2021. Quantum option pricing using Wick rotated imaginary time evolution. *preprint arXiv:2101.04280* (2021).

[204] Sergi Ramos-Calderer, Adrián Pérez-Salinas, Diego García-Martín, Carlos Bravo-Prieto, Jorge Cortada, Jordi Planaguma, and José I. Latorre. 2021. Quantum unary approach to option pricing. *Physical Review A* 103, 3 (2021), 032414.

[205] Anupama Ray, Sai Sakunthala Guddanti, Vishnu Ajith, and Dhinakaran Vinayagamurthy. 2022. Classical ensemble of Quantum-classical ML algorithms for Phishing detection in Ethereum transaction networks. *preprint arXiv:2211.00004* (2022).

[206] Patrick Rebentrost, Brajesh Gupt, and Thomas R. Bromley. 2018. Quantum computational finance: Monte Carlo pricing of financial derivatives. *Physical Review A* 98, 2 (2018), 022321.

[207] Patrick Rebentrost and Seth Lloyd. 2018. Quantum computational finance: quantum algorithm for portfolio optimization. *preprint arXiv:1811.03975* (2018).

[208] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. 2014. Quantum support vector machine for big data classification. *Physical Review Letters* 113, 13 (2014), 130503.

[209] Richard J. Rendleman and Brit J. Bartter. 1979. Two-State Option Pricing. *The Journal of Finance* 34, 5 (1979), 1093–1110. http://www.jstor.org/stable/2327237

[210] Mayra Alejandra Rivera-Ruiz, Andres Mendez-Vazquez, and José Mauricio López-Romero. 2022. Time series forecasting with Quantum Machine Learning architectures. In *Mexican International Conference on Artificial Intelligence*. Springer, 66–82. https://doi.org/10.1007/978-3-031-19493-1_6

[211] Thierry Roncalli. 2013. *Introduction to risk parity and budgeting.* CRC Press.

[212] Gilli Rosenberg. 2016. Finding optimal arbitrage opportunities using a quantum annealer. *1QB Information Technologies Write Paper* (2016), 1–7. arXiv:https://1qbit.com/whitepaper/arbitrage/

[213] Gili Rosenberg, Poya Haghnegahdar, Phil Goddard, Peter Carr, Kesheng Wu, and Marcos López De Prado. 2015. Solving the optimal trading trajectory problem using a quantum annealer. In *Proceedings of the 8th Workshop on High Performance Computational Finance*. 1–7.

[214] Álvaro Rubio-García, Juan José García-Ripoll, and Diego Porras. 2022. Portfolio optimization with discrete simulated annealing. *preprint arXiv:2210.00807* (2022).

[215] Lidia Ruiz-Perez and Juan Carlos Garcia-Escartin. 2017. Quantum arithmetic with the quantum Fourier transform. *Quantum Information Processing* 16 (2017), 1–14.

[216] Alona Sakhnenko, Corey O'Meara, Kumar J. B. Ghosh, Christian B. Mendl, Giorgio Cortiana, and Juan Bernabé-Moreno. 2022. Hybrid classical-quantum autoencoder for anomaly detection. *Quantum Machine Intelligence* 4, 2 (2022), 27.

[217] Maria Schuld, Alex Bocharov, Krysta M. Svore, and Nathan Wiebe. 2020. Circuit-centric quantum classifiers. *Phys. Rev. A* 101 (Mar 2020), 032308. Issue 3. https://doi.org/10.1103/PhysRevA.101.032308

[218] Maria Schuld and Nathan Killoran. 2022. Is Quantum Advantage the Right Goal for Quantum Machine Learning? *PRX Quantum* 3 (Jul 2022), 030101. Issue 3. https://doi.org/10.1103/PRXQuantum.3.030101

[219] Maria Schuld and Francesco Petruccione. 2021. *Machine learning with quantum computers.* Springer.

[220] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. 2016. Prediction by linear regression on a quantum computer. *Physical Review A* 94, 2 (2016), 022342.

[221] Diksha Sharma, Parvinder Singh, and Atul Kumar. 2022. A comparative study of classical and quantum Machine Learning models for sentimental analysis. *preprint arXiv:2209.05142* (2022).

[222] Surbhi Sharma and Baijnath Kaushik. 2018. Quantitative analysis of stock market prediction for accurate investment decisions in future. *Journal of Artificial Intelligence* 11, 1 (2018), 48–54.

[223] William F. Sharpe. 1966. Mutual fund performance. *The Journal of Business* 39, 1 (1966), 119–138.

[224] William F. Sharpe. 1998. The Sharpe ratio. *Streetwise–the Best of the Journal of Portfolio Management* 3 (1998), 169–85.

[225] Albert N. Shiryaev. 2007. *Optimal stopping rules.* Vol. 8. Springer Science & Business Media.

[226] Naman Srivastava, Gaurang Belekar, Neel Shahakar, and Aswath Babu H. 2023. The Potential of Quantum Techniques for Stock Price Prediction. *IEEE International Conference on Recent Advancements in Systems Science and Engineering* (2023). arXiv:arXiv:2308.13642

[227] Nikitas Stamatopoulos, Daniel J. Egger, Yue Sun, Christa Zoufal, Raban Iten, Ning Shen, and Stefan Woerner. 2020. Option pricing using quantum computers. *Quantum* 4 (2020), 291.

[228] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement learning: An introduction.* MIT press.

[229] Yohichi Suzuki, Shumpei Uno, Rudy Raymond, Tomoki Tanaka, Tamiya Onodera, and Naoki Yamamoto. 2020. Amplitude estimation without phase estimation. *Quantum Information Processing* 19 (2020), 1–17.

[230] Yasuhiro Takahashi, Seiichiro Tani, and Noboru Kunihiro. 2010. Quantum addition circuits and unbounded fan-out. *Quantum Inf. Comput.* 10, 9&10 (2010), 872–890. https://doi.org/10.26421/QIC10.9-10-12

[231] Hao Tang, Anurag Pal, Tian-Yu Wang, Lu-Feng Qiao, Jun Gao, and Xian-Min Jin. 2021. Quantum computation for pricing the collateralized debt obligations. *Quantum Engineering* 3, 4 (2021), e84.

[232] Elena Peña Tapia, Giannicola Scarpa, and Alejandro Pozas-Kerstjens. 2022. Fraud detection with a single-qubit quantum neural network. *preprint arXiv:2211.13191* (2022).

[233] Chandana Gouri Tekkali and Karthika Natarajan. 2023. Smart payment fraud detection using QML : A major challenge. In *2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS)*. 523–526. https://doi.org/10.1109/ICAIS56108.2023.10073712

[234] Sohum Thakkar, Skander Kazdaghli, Natansh Mathur, Iordanis Kerenidis, André J. Ferreira-Martins, and Samurai Brito. 2023. Improved financial forecasting via Quantum Machine Learning. *preprint arXiv:2306.12965* (2023).

[235] Himanshu Thapliyal and Nagarajan Ranganathan. 2013. Design of efficient reversible logic-based binary and BCD adder circuits. *J. Emerg. Technol. Comput. Syst.* 9, 3, Article 17 (oct 2013), 31 pages. https://doi.org/10.1145/2491682

[236] Jules Tilly, Hongxiang Chen, Shuxiang Cao, Dario Picozzi, Kanav Setia, Ying Li, Edward Grant, Leonard Wossnig, Ivan Rungger, George H. Booth, et al. 2022. The variational quantum eigensolver: a review of methods and best practices. *Physics Reports* 986 (2022), 1–128.

[237] Joseph Tindall, Matthew Fishman, E. Miles Stoudenmire, and Dries Sels. 2024. Efficient Tensor Network Simulation of IBM's Eagle Kicked Ising Experiment. *PRX Quantum* 5 (Jan 2024), 010308. Issue 1. https://doi.org/10.1103/PRXQuantum.5.010308

[238] Kilian Tscharke, Sebastian Issel, and Pascal Debus. 2023. Semisupervised Anomaly Detection using Support Vector Regression with Quantum Kernel. In *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, Vol. 01. 611–620. https://doi.org/10.1109/QCE57702.2023.00075

[239] Laurens Van Der Maaten, Eric O. Postma, H. Jaap van den Herik, et al. 2009. Dimensionality reduction: A comparative review. *Journal of Machine Learning Research* 10, 66-71 (2009), 13.

[240] Rodney Van Meter and Kohei M. Itoh. 2005. Fast quantum modular exponentiation. *Physical Review A* 71, 5 (2005), 052320.

[241] Vlatko Vedral, Adriano Barenco, and Artur Ekert. 1996. Quantum networks for elementary arithmetic operations. *Physical Review A* 54, 1 (1996), 147.

[242] Davide Venturelli and Alexei Kondratyev. 2019. Reverse quantum annealing approach to portfolio optimization problems. *Quantum Machine Intelligence* 1, 1-2 (2019), 17–30.

[243] Martin Veselý. 2022. Application of quantum computers in Foreign Exchange reserves management. *preprint arXiv:2203.15716* (2022).

[244] David Von Dollen, Florian Neukart, Daniel Weimer, and Thomas Bäck. 2021. Quantum-assisted feature selection for vehicle price prediction modeling. *preprint arXiv:2104.04049* (2021).

[245] Ahmed S. Wafi, Hassan Hassan, and Adel Mabrouk. 2015. Fundamental analysis models in financial markets–review study. *Procedia Economics and Finance* 30 (2015), 939–947.

[246] Abraham Wald. 2004. *Sequential analysis.* Courier Corporation.

[247] Guoming Wang. 2017. Quantum algorithm for linear regression. *Physical Review A* 96, 1 (2017), 012335.

[248] Hongbin Wang and Minghui Hou. 2023. Quantum-like implicit sentiment analysis with sememes knowledge. *Expert Systems with Applications* (2023), 120720.

[249] Hua Wang, Cuiqin Ma, and Lijuan Zhou. 2009. A brief review of machine learning and its application. In *2009 International Conference on Information Engineering and Computer Science*. 1–4. https://doi.org/10.1109/ICIECS.2009.5362936

[250] Haibo Wang, Wendy Wang, Yi Liu, and Bahram Alidaee. 2022. Integrating machine learning algorithms with quantum annealing solvers for online fraud detection. *IEEE Access* 10 (2022), 75908–75917.

[251] LuoChao Wang and Raymond S. T. Lee. 2022. The design and implementation of quantum finance Software Development Kit (QFSDK) for AI education. In *2022 20th International Conference on Information Technology Based Higher Education and Training (ITHET)*. IEEE, 1–7.

[252] Maida Wang, Anqi Huang, Yong Liu, Xuming Yi, Junjie Wu, and Siqi Wang. 2023. A quantum-classical hybrid solution for deep anomaly detection. *Entropy* 25, 3 (2023). https://doi.org/10.3390/e25030427

[253] Jarrod West and Maumita Bhattacharya. 2016. Intelligent financial fraud detection: a comprehensive review. *Computers & Security* 57 (2016), 47–66.

[254] Nathan Wiebe, Daniel Braun, and Seth Lloyd. 2012. Quantum algorithm for data fitting. *Physical Review Letters* 109, 5 (2012), 050505.

[255] Peter Wittek. 2014. *Quantum machine learning: what quantum computing means to data mining.* Academic Press.

[256] Stefan Woerner and Daniel J. Egger. 2019. Quantum risk analysis. *npj Quantum Information* 5, 1 (2019), 15.

[257] Li Xie and Sheng Yu. 2021. Unsupervised feature extraction with convolutional autoencoder with application to daily stock market prediction. *Concurrency and Computation: Practice and Experience* 33, 16 (2021), e6282.

[258] Romina Yalovetzky, Pierre Minssen, Dylan Herman, and Marco Pistoia. 2021. NISQ-HHL: Portfolio optimization for near-term quantum hardware. *preprint arXiv:2110.15958* (2021).

[259] Özlem Yeşiltaş. 2023. The Black–Scholes equation in finance: Quantum mechanical approaches. *Physica A: Statistical Mechanics and its Applications* (2023), 128909.

[260] P. D. Yoo, M. H. Kim, and T. Jan. 2005. Machine Learning Techniques and Use of Event Information for Stock Market Prediction: A Survey and Evaluation. In *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, Vol. 2. 835–841. https://doi.org/10.1109/CIMCA.2005.1631572

[261] Rizgar Zebari, Adnan Abdulazeez, Diyar Zeebaree, Dilovan Zebari, and Jwan Saeed. 2020. A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction. *Journal of Applied Science and Technology Trends* 1, 2 (2020), 56–70.

[262] Carlos Vladimiro González Zelaya. 2019. Towards explaining the effects of data preprocessing on machine learning. In *2019 IEEE 35th International Conference on data engineering (ICDE)*. IEEE, 2086–2090.

[263] Leo Zhou, Sheng-Tao Wang, Soonwon Choi, Hannes Pichler, and Mikhail D. Lukin. 2020. Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices. *Physical Review X* 10, 2 (2020), 021067.

[264] SiSi Zhou, Thomas Loke, Josh A. Izaac, and J. B. Wang. 2017. Quantum Fourier transform in computational basis. *Quantum Information Processing* 16, 3 (2017), 82.

[265] Christa Zoufal, Aurélien Lucchi, and Stefan Woerner. 2019. Quantum generative adversarial networks for learning and loading random distributions. *npj Quantum Information* 5, 1 (2019), 103.

[266] Christa Zoufal, Ryan V. Mishmash, Nitin Sharma, Niraj Kumar, Aashish Sheshadri, Amol Deshmukh, Noelle Ibrahim, Julien Gacon, and Stefan Woerner. 2023. Variational quantum algorithm for unconstrained black box binary optimization: Application to feature selection. *Quantum* 7 (Jan. 2023), 909. https://doi.org/10.22331/q-2023-01-26-909