Quantum Algorithms for Trust-Based AI Applications

Davinder Kaur, Suleyman Uslu, and Arjan Durresi

Abstract Quantum computing is a rapidly growing field of computing that leverages the principles of quantum mechanics to significantly speed up computations that are beyond the capabilities of classical computing. This type of computing can revolutionize the field of trustworthy artificial intelligence, where decision-making is data-driven, complex, and time-consuming. Different trust-based AI systems have been proposed for different AI applications. In this paper, we have reviewed different trust-based AI systems and summarized their alternative quantum algorithms. This review provides an overview of quantum algorithms for three trust-based AI applications: fake user detection in social networks, medical diagnostic system, and finding the shortest path used in social network trust aggregation.

1 Introduction

Quantum Computing is the field that uses quantum mechanical phenomena such as superposition and entanglement to perform operations on data much more efficiently than classical computing. It is the intersection of physics, mathematics, and computer science. Quantum computing can perform many computations simultaneously. This computing technology is based on qubits, which can exist in multiple states simultaneously. It provides several advantages over the classical computing methods because it drastically reduces the execution time and energy consumption [37].

Davinder Kaur
Indiana University-Purdue University Indianapolis, Indianapolis, IN, USA e-mail: davikaur@iu.edu

Suleyman Uslu
Indiana University-Purdue University Indianapolis, Indianapolis, IN, USA e-mail: suslu@iu.edu

Arjan Durresi
Indiana University-Purdue University Indianapolis, Indianapolis, IN, USA e-mail: adurresi@iupui.edu

Quantum computing is an innovative and life-changing technology. Recently, Google has been investing billions of dollars in building its quantum computer by 2029 [1]. IBM and Microsoft are also working on providing quantum computing benefits to customers [25]. This technology can solve complex problems that are difficult to solve using classical computing. Inspired by the help of quantum computing, in this paper, we have reviewed the quantum algorithms for some of the real-world trust-based artificial intelligence applications that are generally time-consuming using classical algorithms. The trust-based AI applications are widely used in our day-to-day lives. It is essential to make these applications safe, reliable, and trustworthy [16] [20] by integrating AI applications with trust assessment techniques [31, 33]. Different researchers have proposed trust-based methods to create various AI applications trustworthy. Some researchers proposed a trust framework for AI applications in Food, Energy, and Water sectors [38, 39, 40, 41, 42, 43], some proposed for fake user detection [19][30], and some suggested for medical diagnostics systems [18][17]. So, in this review, we have discussed the quantum alternatives of some widely used algorithms for trust-based AI systems.

This paper is organized as follows. Section 2 presents the background of quantum computing. Section 3 reviews the quantum algorithms for widely used trust-based AI applications. And in Section 4, we conclude the paper.

2 Background

Quantum Computing is a new paradigm that leverages the concept of quantum mechanics to process information differently from classical computing approaches. Quantum mechanics explains the behavior of the particles on the quantum level, i.e., sub-atomic and atomic levels [35]. This type of computing method is used to solve complex problems which are challenging to solve using traditional approaches. A quantum computer consists of various components, including:

- Qubit: Qubit is the building block of quantum computers. It is the fundamental information-carrying unit. It is a quantum analog of classical bits and can exist in multiple states simultaneously. More precisely, classical computers use binary bits: 0s and 1s, whereas quantum computer uses 0s, 1s, and both a 0 and 1 simultaneously. The capability of having multiple states at the same time gives quantum computers immense processing power.
- Superposition: Superposition is the ability of a qubit to exist in multiple states simultaneously. It refers to the linear combination of two quantum states. In quantum computing, a qubit can exist as a superposition of two states(0 and 1) and can perform multiple computations simultaneously.
- Entanglement: Entanglement is the phenomenon in which two or more qubits become correlated so that their states become dependent on each other. Quantum entanglement enables qubits separated by large distances to interact with each other instantaneously. This property type enables one to perform certain types of computations more efficiently.

- Quantum Gates: Quantum gates are basic building blocks for quantum circuits. They are similar to logic gates used in classical computing. These gates operate on qubits to manipulate their states to perform complex computations. These gates perform operations like superposition, entanglement, and measurements.
- Quantum Memory: Quantum memory stores quantum information, which is fragile and can be easily lost due to environmental disturbances. Several types of quantum memory exist, including superconducting qubits, trapped ions, and topological qubits.
- Quantum Algorithms: Quantum algorithms are a set of algorithms that take advantage of the unique properties of quantum computers to perform specific calculations more efficiently than classical computers [1].

3 Review

This section provides a review of three trust-based AI applications, their widely used classical algorithms, and the quantum alternative of those algorithms.

3.1 Fake User Detection in Social Networks

Social networks have become an integral part of people's daily lives. However, with the increased use of social networks also comes certain risks, such as the spread of fake news, malicious content, and viruses by creating fake accounts [15]. It is essential to detect these fake user accounts as soon as possible and take action to prevent the spread of harmful content. Detecting fake user clusters in social networks is challenging, as fake users can use various tactics to evade detection. However, several methods can be used to detect fake user clusters in social networks. Some researchers proposed analyzing user profiles and behavior to detect fake users [45]. Other researchers proposed methods that utilize the graphical properties of social networks to detect fake users [6]. Another set of researchers uses trust information between users to detect malicious and fake users [19][30].

Combining all these approaches for clustering social network graphs is an effective way to detect clusters of fake users in social networks. The traditional clustering algorithms on large social network graphs are computationally expensive and time-consuming. Quantum computers and quantum clustering algorithms provide exponential speed-ups to the conventional clustering approaches. Following subsections explain the traditional clustering algorithms and their quantum alternatives.

3.1.1 Traditional DBSCAN Clustering Approach

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is an unsupervised machine learning method to find arbitrary shape clusters and clusters with noise. This algorithm group together the points that are close to each other in terms of distance and density [11]. The main idea behind this clustering technique is that if a point is close to many points from a cluster, it belongs to that cluster. It takes two parameters as input: epsilon and minPts. Epsilon is a distance threshold that defines the radius of a neighborhood around each point, and minPts is the minimum number of points to define a cluster. The algorithm starts by selecting the random unvisited point, and its neighborhood is determined using epsilon. If the neighborhood has at least minPts, cluster formation starts. The algorithm expands recursively. In the next step, the algorithm chooses another point that has not been visited in the previous step, and the process continues until all the points have been visited [9].

The advantages of DBSCAN include its ability to find clusters of arbitrary shapes and sizes, its tolerance for noise points, and its ability to handle datasets with varying densities. However, the algorithm can be sensitive to the choice of epsilon and minPts, may not perform well on datasets with different local density clusters, and is time-consuming.

3.1.2 Quantum DBSCAN Clustering Algorithm

Inspired by the advantages of quantum computing, [46] proposed a quantum Mutual MinPts-nearest Neighbor Graph (MMNG) based DBSCAN algorithm. This algorithm performs better on datasets with different low-density clusters and dramatically increases speed compared to the traditional approach. The proposed algorithm comprises two sub-algorithms: a Quantum mutual MinPts-nearest neighbor graph algorithm and a quantum DBSCAN algorithm. The Quantum mutual MinPts-nearest neighbor graph algorithm divides the dataset into subsets. And on each of the generated subsets, the quantum DBSCAN algorithm is applied to obtain clusters and noise set. Different subsets have different epsilon for this algorithm. In the quantum DBSCAN algorithm, the distance calculation needed to determine the Epsilon neighborhood is done using quantum search. The steps of the Quantum MMNG DBSCAN algorithm are given below:

Algorithm 1 Quantum MMNG DBSCAN

Input: Dataset, minPts
Output: Cluster and noise set
Procedure:
Step 1: Divide the dataset into subsets using the quantum - MMNG algorithm [46].
Step 2: For every subset obtained in Step 1, calculate the epsilon and get clusters and noise using the quantum DBSCAN algorithm [46].
Step 3: Return all the clusters and noise set.

The complexity of the proposed algorithm is O($N\sqrt{minPts*n}$), where n is the number of data points, and minPts is the minimum number of data points required to define the cluster.

3.1.3 Louvain Community Detection Algorithm

Louvain algorithm is an unsupervised community detection algorithm used to detect communities from large networks [7]. This algorithm does not require the user input of community size or the number of communities before execution. The algorithm comprises the modularity optimization phase and the community aggregation phase [3]. In the modularity optimization phase, each node is assigned to its community, and the algorithm iteratively evaluates the modularity gain resulting from merging nodes with its neighbors. Only the nodes that result in the highest modularity gain will be moved to the community. In the second phase, communities detected in the first phase are aggregated, and the first phase is repeated on this new network. This process is repeated until no further modularity gain can be achieved. Figure 1 shows different social network communities detected by the Louvain algorithm in detecting fake users [19].

Louvain algorithm has several advantages like speed, scalability, and flexibility, making it suitable for detecting communities in large social network graphs [3]. The quantum variant of this algorithm provides quantum speedups to the task of community detection in large complex networks [5]. The following section discusses the quantum variant of the Louvain community detection algorithm.

3.1.4 Quantum Variant of Louvain Algorithm

The Quantum variant of the Louvain algorithm is the EdgeQLouvain. This algorithm utilizes a single Grover search over an ample search space (the entire set of vertices) rather than searching over vertices and their neighbors [5] to find a good move. Given the input graph with directed edges (u,v), this algorithm searches for the edge that will increase the modularity if u is moved to the neighboring community. This version of the quantum Louvain algorithm has several advantages as it does not need nested Grover search, making it much simplified and faster than other quantum variants [5]. Given an edge set E of the graph that contains undirected edges, a directed graph is obtained by replacing every undirected edge {u,v} with (u,v) and (v,u), and on this directed graph, the algorithm as described below is applied.

This algorithm shows polynomial speedup as compared to the traditional Louvain algorithm. The query complexity of this algorithm for every step k is O($1/\sqrt{h_k}$) where h_k is the fraction of the edges.

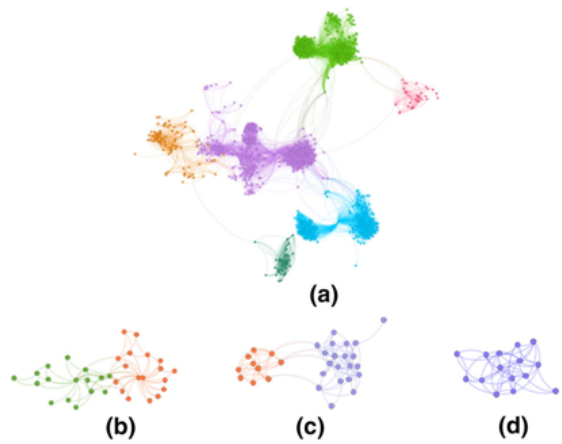

Fig. 1 Communities detected by the Louvain algorithm for different social networks. a) Facebook ego network. b) Karate friends network. c) Fake user network 1 and d) Fake User Network 2

Algorithm 2 Edge Quantum Louvain Algorithm

Input: Graph edge set
Output: Cluster set
Procedure:
Step 1: Initially, assign every vertex to its community.
Step 2: Utilize the quantum algorithm QSearch [4] to search all over the edges(u,v) to find one that yields a good move.
Step 3: Find the best neighboring community of u using the quantum maximum finding algorithm [8].
Step 4: Repeat steps 2 and 3 until there is no modularity increase.
Step 5: Aggregate communities to make a new graph and repeat steps 2-4 until there is no more change.

3.2 AI system for Medical Diagnostics

Artificial intelligence and machine learning systems have completely changed our lives. Many high stake applications like medical diagnostics are widely adopting these systems. With the vast amount of data and computing power available, these algorithms have become very good at predicting diagnostic results and saving time and money [36]. These algorithms are helping doctors with a cancer diagnosis by analyzing the image dataset of old cancer cases by detecting, measuring, and exploring the tumor cells [18][17].

The widely used machine learning algorithms in medical diagnostics are classification algorithms. The classification algorithms are the supervised algorithms that take training data as the input to predict the likelihood or probability of the new data to belong into the predetermined categories. To speed up the classical machine learning classification algorithms, quantum machine learning is introduced, integrating the quantum algorithms with the classification algorithms [2]. The following subsection discusses the traditional classification algorithms and their quantum alternatives.

3.2.1 Traditional Support Vector Machine

Support vector machine (SVM) is a supervised machine learning algorithm for classification and regression tasks. This algorithm is widely used in biological applications as it learns to assign labels by learning from examples [27]. SVM works by finding the best possible boundary or hyperplane that distinctly classifies the data points of different classes in the high dimensional space. This algorithm aims to find a hyperplane that maximizes the margin, that is, the distance between the hyperplane and the closest data points from each class. This algorithm is helpful in high-dimensional spaces as it uses kernel functions to transform the input data into high-dimensional space where it is easier to find a separating hyperplane.

This algorithm has several advantages, including its robustness to the outliers and effectiveness in cases where the number of samples is lesser than the number of features. However, this algorithm can be computationally intensive and not suitable for large data sets [44].

3.2.2 Quantum Support Vector Machine

The quantum support vector machine algorithm performs the least square-SVM using a quantum computer [29]. This quantum algorithm uses phase estimation and a quantum matrix inversion algorithm to maximize the algorithm's speed. Considering there are N data points (x_i, y_i): i = 1, 2.....N), where x_i is the feature vector, and y_i is the binary label of the data, the goal of the SVM is to find the hyperplane $w.x + b = 0$ that divides the data points into two categories. The quantum least square SVM algorithm calculates the kernel matrix using the quantum random access memory [10] and solves the linear equation using the quantum algorithm for solving linear equations and then performs classification using the trained qubits. Following are the steps of the quantum support vector machine algorithm.

Algorithm 3 Quantum Support Vector Machine

Input: Training Data and Test Data
Output: Classification: +1 or -1
Procedure:
Step 1: Calculate the kernel metrics using the quantum inner product algorithm [24].
Step 2: Solve the linear equation using the quantum algorithm for linear equations [10].
Step 3: Classify the test data using the training results, using a quantum algorithm [29].

The complexity of the quantum SVM algorithm is $O(logNM)$ as compared to the traditional SVM, which has the complexity of $O(M^2(M+N))$. Here N is the N-dimensional feature vector, and M is the number of data points.

3.2.3 Traditional Logistic Regression

Logistic regression is a supervised machine learning algorithm widely used for classification tasks. This algorithm classifies observations into a discrete set of classes. Logistic regression predicts the probability of an event occurring based on a set of independent variables or predictors. The algorithm models the relationship between the dependent variable and one or more independent variables using a logistic function, which produces an S-shaped curve [21]. This logistic function transforms the linear combination of input variables and coefficients into a probability value of 0 and 1. The logistic regression algorithm minimizes the difference between the predicted probabilities and the actual output in the training data by finding the optimal values for the coefficients.

Logistic regression is widely used in many fields, including healthcare, finance, and social sciences. It has many advantages: it can handle categorical and continuous variables as input features. However, this algorithm does not perform well if there is a non-linear relationship between the input and output variables and is time-consuming when applied to big datasets.

3.2.4 Quantum Logistic Regression

Logistic regression is an important algorithm used for classification tasks. But this algorithm can be slow for large data sets as it involves a gradient descent method at each iteration which is quite time-consuming. To overcome this, [23] proposed a quantum logistic regression algorithm that implements the critical task of the gradient descent at each iteration, making the algorithm exponentially faster than the classical logistic regression. The quantum algorithm is divided into two steps: the first is to generate the quantum state using the amplitude estimation [4], and the second step is using the swap test [34] to obtain a gradient in the classical form. To ensure the new data can be classified by this algorithm directly, it outputs the model parameters in classical form. Following are the steps for the quantum logistic regression algorithm.

The quantum logistic regression algorithm provides exponential speed compared to the traditional algorithm. The complexity of this algorithm is $O(polylogN)$ for every iteration, where N is the number of data points.

Algorithm 4 Quantum Logistic Regression

Input: Training Data and Test Data
Output: Classification: +1 or -1
Procedure:
Step 1: Initialize all parameters.
Step 2: Calculate the dependent variable.
Step 3: Calculate the cost function.
Step 4: Calculate the gradient of the cost function using a quantum algorithm that consists of amplitude estimation [4] and swap test [34].
Step 5: Update all parameters.
Step 6: Repeat steps 2-5.

3.3 Finding Shortest Path

Shortest path algorithms are used to find the shortest path between points. Given a graph, these algorithms find the shortest path from one point to another or from one point to all other points [26]. These algorithms have many applications. In traffic information systems, these algorithms are used to find the optimal path from one point to another point. In networking, these algorithms are used in routing protocols to find the optimal path to transmit data packets. These algorithms are also used in social network analysis and autonomous vehicle route planning. For example, the highest confidence path can be used to speed up calculations in stock market prediction using Twitter trust networks [32].

To make the process of finding the shortest path faster, quantum algorithms for finding the shortest path are proposed as they are capable of performing several operations simultaneously [22]. The following subsections discuss the widely used traditional shortest path algorithm and its quantum alternative.

3.3.1 Traditional Dijkstra Algorithm

Dijkstra algorithm is a popular algorithm for solving the single source shortest path search in the weighted graphs with non-negative weights [13]. This algorithm finds the shortest path from the source node to all other nodes. This algorithm is handy in the traffic information system to find the shortest path between the current location and the destination and also in modeling networks.

The algorithm keeps track of visited and non-visited nodes. Initially, it starts with the source node, whose distance is zero. Then for each non-visited neighbor of the current node, the algorithm calculates the distance from the source node to that neighbor by using the weights of the edges connecting them. If the distance is less than the current distance, it updates it. This process is repeated until all the nodes have been visited [12]. This way, the algorithm finds the shortest path between the source and any other node in the graph.

3.3.2 Quantum Dijkstra Algorithm

Researchers [28] have proposed the quantum version of the Dijkstra algorithm that utilizes the principles of quantum superposition and inference to find the shortest path in the graph. This algorithm proves to be better in terms of time complexity than the traditional Dijkstra algorithm. This algorithm utilizes the quantum search algorithm and phase estimation to speed up the search operation. The following steps explain the quantum Dijkstra algorithm:

Algorithm 5 Quantum Dijkstra Algorithm

Input: Set of nodes, source node.
Output: The shortest path from the source node to all the other nodes.
Procedure:
Step 1: Initialize the distance to the source node to zero, and all other nodes to infinity.
Step 2: Create a set of visited and unvisited nodes.
Step 3: While there are unvisited nodes:
Select the smallest distance unvisited neighbor and find the minimum distance path from the source node to the neighbor using Grover's algorithm [14] and quantum minimum searching algorithm [8].
Step 4: Output the best path from the source to all the nodes.

The complexity of this algorithm is $O(\sqrt{NM}log^2N)$ as compared to the traditional Dijkstra algorithm, which has the complexity of $O(M+NlogN)$. Here N is the number of vertexes, and M is the number of edges.

4 Conclusion

Quantum computing provides several benefits as compared to classical computing. Different researchers have proposed different quantum algorithms that offer significant benefits. However, there was a lack of mapping between the quantum algorithms and the real-life applications. This review summarizes quantum algorithms for trust-based AI applications for fake user detection, medical diagnostics, and finding the shortest path in trust networks.

References

1. Adedoyin, A., Ambrosiano, J., Anisimov, P., Bärtschi, A., Casper, W., Chennupati, G., Coffrin, C., Djidjev, H., Gunter, D., Karra, S., et al.: Quantum algorithm implementations for beginners. arXiv preprint arXiv:1804.03719 (2018)
2. Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., Lloyd, S.: Quantum machine learning. Nature **549**(7671), 195–202 (2017)
3. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of statistical mechanics: theory and experiment **2008**(10), P10,008 (2008)

4. Brassard, G., Hoyer, P., Mosca, M., Tapp, A.: Quantum amplitude amplification and estimation. Contemporary Mathematics **305**, 53–74 (2002)
5. Cade, C., Folkertsma, M., Niesen, I., Weggemans, J.: Quantum algorithms for community detection and their empirical run-times. arXiv preprint arXiv:2203.06208 (2022)
6. Cao, Q., Sirivianos, M., Yang, X., Pregueiro, T.: Aiding the detection of fake accounts in large scale social online services. In: Presented as part of the 9th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 12), pp. 197–210 (2012)
7. De Meo, P., Ferrara, E., Fiumara, G., Provetti, A.: Generalized louvain method for community detection in large networks. In: 2011 11th international conference on intelligent systems design and applications, pp. 88–93. IEEE (2011)
8. Durr, C., Hoyer, P.: A quantum algorithm for finding the minimum. arXiv preprint quant-ph/9607014 (1996)
9. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: kdd, vol. 96, pp. 226–231 (1996)
10. Giovannetti, V., Lloyd, S., Maccone, L.: Quantum random access memory. Physical review letters **100**(16), 160,501 (2008)
11. Hinneburg, A.: A density based algorithm for discovering clusters in large spatial databases with noise. In: KDD Conference, 1996 (1996)
12. Javaid, A.: Understanding dijkstra's algorithm. Available at SSRN 2340905 (2013)
13. Johnson, D.B.: A note on dijkstra's shortest path algorithm. Journal of the ACM (JACM) **20**(3), 385–388 (1973)
14. Jozsa, R.: Searching in grover's algorithm. arXiv preprint quant-ph/9901021 (1999)
15. Kaur, D., Uslu, S., Durresi, A.: Trust-based security mechanism for detecting clusters of fake users in social networks. In: Web, Artificial Intelligence and Network Applications: Proceedings of the Workshops of the 33rd International Conference on Advanced Information Networking and Applications (WAINA-2019) 33, pp. 641–650. Springer (2019)
16. Kaur, D., Uslu, S., Durresi, A.: Requirements for trustworthy artificial intelligence–a review. In: Advances in Networked-Based Information Systems: The 23rd International Conference on Network-Based Information Systems (NBiS-2020) 23, pp. 105–115. Springer (2021)
17. Kaur, D., Uslu, S., Durresi, A.: Trustworthy ai explanations as an interface in medical diagnostic systems. In: Advances in Network-Based Information Systems: The 25th International Conference on Network-Based Information Systems (NBiS-2022), pp. 119–130. Springer (2022)
18. Kaur, D., Uslu, S., Durresi, A., Badve, S., Dundar, M.: Trustworthy explainability acceptance: A new metric to measure the trustworthiness of interpretable ai medical diagnostic systems. In: Complex, Intelligent and Software Intensive Systems: Proceedings of the 15th International Conference on Complex, Intelligent and Software Intensive Systems (CISIS-2021), pp. 35–46. Springer (2021)
19. Kaur, D., Uslu, S., Durresi, M., Durresi, A.: A geo-location and trust-based framework with community detection algorithms to filter attackers in 5g social networks. Wireless Networks pp. 1–9 (2022)
20. Kaur, D., Uslu, S., Rittichier, K.J., Durresi, A.: Trustworthy artificial intelligence: a review. ACM Computing Surveys (CSUR) **55**(2), 1–38 (2022)
21. Kleinbaum, D.G., Dietz, K., Gail, M., Klein, M., Klein, M.: Logistic regression. Springer (2002)
22. Krauss, T., McCollum, J.: Solving the network shortest path problem on a quantum annealer. IEEE Transactions on Quantum Engineering **1**, 1–12 (2020)
23. Liu, H.L., Yu, C.H., Wu, Y.S., Pan, S.J., Qin, S.J., Gao, F., Wen, Q.Y.: Quantum algorithm for logistic regression. arXiv preprint arXiv:1906.03834 (2019)
24. Lloyd, S., Mohseni, M., Rebentrost, P.: Quantum algorithms for supervised and unsupervised machine learning. arXiv preprint arXiv:1307.0411 (2013)
25. MacQuarrie, E.R., Simon, C., Simmons, S., Maine, E.: The emerging commercial landscape of quantum computing. Nature Reviews Physics **2**(11), 596–598 (2020)
26. Magzhan, K., Jani, H.M.: A review and evaluations of shortest path algorithms. International journal of scientific & technology research **2**(6), 99–104 (2013)

27. Noble, W.S.: What is a support vector machine? Nature biotechnology **24**(12), 1565–1567 (2006)
28. Ray, P.: Quantum simulation of dijkstra's algorithm. Int. J. Adv. Res. Comput. Sci. Manag. Stud **2**, 30–43 (2014)
29. Rebentrost, P., Mohseni, M., Lloyd, S.: Quantum support vector machine for big data classification. Physical review letters **113**(13), 130,503 (2014)
30. Rittichier, K.J., Kaur, D., Uslu, S., Durresi, A.: A trust-based tool for detecting potentially damaging users in social networks. In: Advances in Networked-Based Information Systems: The 24th International Conference on Network-Based Information Systems (NBiS-2021), pp. 94–104. Springer (2022)
31. Ruan, Y., Durresi, A.: A survey of trust management systems for online social communities–trust modeling, trust inference and attacks. Knowledge-Based Systems **106**, 150–163 (2016)
32. Ruan, Y., Durresi, A., Alfantoukh, L.: Using twitter trust network for stock market analysis. Knowledge-Based Systems **145**, 207–218 (2018)
33. Ruan, Y., Zhang, P., Alfantoukh, L., Durresi, A.: Measurement theory-based trust management framework for online social communities. ACM Transactions on Internet Technology (TOIT) **17**(2), 16 (2017)
34. Schuld, M., Sinayskiy, I., Petruccione, F.: Prediction by linear regression on a quantum computer. Physical Review A **94**(2), 022,342 (2016)
35. National Academies of Sciences, E., Medicine, et al.: Quantum computing: progress and prospects (2019)
36. Sidey-Gibbons, J.A., Sidey-Gibbons, C.J.: Machine learning in medicine: a practical introduction. BMC medical research methodology **19**, 1–18 (2019)
37. Steane, A.: Quantum computing. Reports on Progress in Physics **61**(2), 117 (1998)
38. Uslu, S., Kaur, D., Rivera, S.J., Durresi, A., Babbar-Sebens, M.: Decision support system using trust planning among food-energy-water actors. In: International Conference on Advanced Information Networking and Applications, pp. 1169–1180. Springer (2019)
39. Uslu, S., Kaur, D., Rivera, S.J., Durresi, A., Babbar-Sebens, M.: Trust-based game-theoretical decision making for food-energy-water management. In: International Conference on Broadband and Wireless Computing, Communication and Applications, pp. 125–136. Springer (2019)
40. Uslu, S., Kaur, D., Rivera, S.J., Durresi, A., Babbar-Sebens, M.: Trust-based decision making for food-energy-water actors. In: International Conference on Advanced Information Networking and Applications, pp. 591–602. Springer (2020)
41. Uslu, S., Kaur, D., Rivera, S.J., Durresi, A., Babbar-Sebens, M., Tilt, J.H.: Control theoretical modeling of trust-based decision making in food-energy-water management. In: Conference on Complex, Intelligent, and Software Intensive Systems, pp. 97–107. Springer (2020)
42. Uslu, S., Kaur, D., Rivera, S.J., Durresi, A., Babbar-Sebens, M., Tilt, J.H.: A trustworthy human–machine framework for collective decision making in food–energy–water management: The role of trust sensitivity. Knowledge-Based Systems **213**, 106,683 (2021)
43. Uslu, S., Kaur, D., Rivera, S.J., Durresi, A., Durresi, M., Babbar-Sebens, M.: Trustworthy acceptance: A new metric for trustworthy artificial intelligence used in decision making in food–energy–water sectors. In: International Conference on Advanced Information Networking and Applications, pp. 208–219. Springer (2021)
44. Wang, L.: Support vector machines: theory and applications, vol. 177. Springer Science & Business Media (2005)
45. Xiao, C., Freeman, D.M., Hwa, T.: Detecting clusters of fake accounts in online social networks. In: Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security, pp. 91–101 (2015)
46. Xie, X., Duan, L., Qiu, T., Li, J.: Quantum algorithm for mmng-based dbscan. Scientific Reports **11**(1), 15,559 (2021)