A Review of k-NN Algorithm Based on Classical and Quantum Machine Learning

Yeray Mezquita[1(✉)], Ricardo S. Alonso[1], Roberto Casado-Vara[1], Javier Prieto[1,2], and Juan Manuel Corchado[1,2]

[1] BISITE Research Group, University of Salamanca, Salamanca, Spain
{yeraymm,ralorin,rober,javierp,corchado}@usal.es
[2] AIR Institute, IoT Digital Innovation Hub, Salamanca, Spain
https://bisite.usal.es, https://www.innovationhub.es

Abstract. Artificial intelligence algorithms, developed for traditional computing, based on Von Neumann's architecture, are slow and expensive in terms of computational resources. Quantum mechanics has opened up a new world of possibilities within this field, since, thanks to the basic properties of a quantum computer, a great degree of parallelism can be achieved in the execution of the quantum version of machine learning algorithms. In this paper, a study has been carried out on these properties and on the design of their quantum computing versions. More specifically, the study has been focused on the quantum version of the k-NN algorithm that allows to understand the fundamentals when transcribing classical machine learning algorithms into its quantum versions.

Keywords: Machine learning · Supervised learning · k-Nearest Neighbors · Quantum computing · Quantum k-NN

1 Introduction

Quantum mechanics is a theory in physics which describes nature at the smallest scales, including atomic and subatomic, and their interactions, as observable quantities. This theory arose to explain observations which could not be properly explained using classical physics. Some examples of first usage of concepts nowadays included in quantum mechanics include the Max Planck's solution to the ultraviolet catastrophe of the black body radiation [29] and the Albert Einstein's explanation of the photoelectric effect [7]. These results lead to suggest the hypothesis of the quantization of energy and that the interaction of an electromagnetic wave with matter occurs by means of elementary and indivisible processes.

In quantum mechanics, the state of a system at a given time is described by a complex wave function, or state vector in a complex vector space. This allows for

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2021
S. Rodríguez González et al. (Eds.): DCAI 2020, AISC 1242, pp. 189–198, 2021.
https://doi.org/10.1007/978-3-030-53829-3_20

computing the probability of finding an electron in a particular region around the nucleus at a particular time. However, it cannot make a simultaneous prediction of related variables, such as position and momentum, to any precision. This is known as the Heisenberg's uncertainty principle which stated the inability to precisely locate the particle given its momentum and conversely [17].

The quantum mechanics has influenced other disciplines, such as chemistry, optics, electronics and information science. Richard Feynman and other authors have proposed that a quantum computer could perform simulations that are out of reach for regular computers [9]. The complexity of the algorithms can be different when executed in a quantum computer. Both problems, integer factorization and discrete log [42], are in polynomial complexity when using quantum computation, however, both are NP problems when using classical computation (although are suspected to not be NP-complete). Maybe because of the heavy computational load it needs, quantum computing is a emerging research area in the discipline of artificial intelligence and specifically applied to the machine learning [38].

This paper is organized as follows. Section 2 describes classical machine learning approaches and algorithms. Section 3 introduces background and terminology about the quantum computing and describe the mathematical background underlying in quantum physics and quantum computing, as well as the improvements obtained by applying quantum computing theories to machine learning algorithms and processes. Section 4 details how the K-Nearest Neighbors (KNN) algorithm is adapted to a quantum version. Finally, conclusions and future work are depicted in Sect. 5.

2 k-NN Algorithm Based on Classical Machine Learning

Artificial Intelligence (AI) aims to develop and use computer systems to reproduce the processes of human intelligence, necessary for learning, understanding, problem solving or decision making. AI is therefore a broad discipline that brings together several fields such as *natural language processing*, *expert systems* [8], *multi-agent systems* [1,43], *recommender systems* [5,12,14], voice analysis and conversion to text (*speech to text* and *text to speech*) [22], *computer vision* [34], *planning systems* [30], *evolutionary computation*, *robotics* and, in the case of our study, *machine learning* [3,4].

According to the different definitions of Artificial Intelligence by [36], *machine learning* falls into those definitions related to *thinking humanly* (related to cognitive systems). Thus, *machine learning* refers to the construction of computer programs that automatically improve their performance in a given task with experience. In machine learning, algorithms are used to parse and learn from data (*training data*). After that, the algorithms make predictions and make decisions about events in the real world. Machine learning algorithms can be subdivided into five main categories: *supervised learning*, *unsupervised learning*, *semi-supervised learning*, *ensemble learning* (also known as *integrated learning*), *deep learning* and *reinforcement learning*, although classification varies over time depending on the importance of each of the groups.

2.1 Supervised Learning

In the case of supervised learning, the machine has a supervisor who indicates an algorithm the result that should be obtained as output for a set of input data. That is, the machine learns about a *labeled dataset*. After learning, the algorithm is able to predict the output for a new combination of input data [40]. Thus, the basic objective of supervised learning is, given a relationship between input variables $\mathbf{X}$ and output variables $\mathbf{Y}$, to learn an objective function f that better maps these input variables to the output variables.

The usual in supervised learning will be that we have a labeled dataset and that we split that dataset in a *training dataset* so that our algorithm finds the relationship f and in a *test dataset* (eliminating the labels) to check the goodness of the function we have obtained with the selected algorithm (i.e., *cross-validation*). An example would be that in a set of image data we indicate for each one which ones correspond to "cats" and which ones correspond to "dog". In this kind of case, where we expect to get a category on the way out, we are talking about *classification* problems. If, on the other hand, we expect to get a continuous numerical value at the output, for example, if we want to predict the value of an action over time, we are talking about a *regression* problem [13].

Regression algorithms are used to predict continuous numerical values rather than discrete variables or categories. This allows us to apply it to predict the value of shares, predict demand or sales volumes, such as electricity consumption [13], make medical diagnoses, and in general, any prediction in time series. Depending on the form of the function found, we find methods of *linear regression*, *polynomial regression* or more modern methods such as *lasso regression* or *ridge regression* (*Tikhonov regularization*) [15].

Classification algorithms are now widely used for applications such as spam filtering, sentiment analysis, language and similar document detection, handwritten character recognition, fraud detection or loan/risk evaluation. Among the most commonly used algorithms are *logistic regression* [31]; *decision trees* (such as ID3, CART, C4.5 or MDL), used in the diagnosis of diseases in medicine or in determining the granting of a loan, among many other possibilities [27]; *classification rules* techniques, such as PRISM; *Naive Bayes classifiers* *li2016convergence*; *Support Vector Machines* [6]; or, focusing on the research of this work, the *K-Nearest Neighbors* algorithm [18].

k-Nearest Neighbors (k-NN) Algorithm. Like decision trees, the *k-NN (k-Nearest Neighbors)* [18] algorithm falls into what are known as *non-parametric learners*. That is, unlike *parametric learners*, this type of method does not require a predefined parametric function $\mathbf{Y} = f(\mathbf{X})$. This makes this type of algorithm suitable for those situations where the relationship between $\mathbf{X}$ and $\mathbf{Y}$ is too complex to be expressed as a linear model. In the k-NN algorithm, each instance is represented as a vector and to classify or make a prediction on an input data, the closest k points are taken and the average of their values is calculated, if we are working with continuous data, like the estimated value of a house, or its mode, if we are working with categorical data, like determining the breed

of a dog. The selection of the k hyperparameter is done by cross-validation, choosing the k that has the least error, on average, along the different iterations. This algorithm is used as a method for classification, as a fraud detection, as a method for regression, as a house price prediction, or to impute missing training data, imputing the average or mode of the neighbors instead of a missing value [19].

3 Quantum Computing in Machine Learning

Although it has not yet been possible to develop commercial hardware for the creation of quantum computers, we can design, albeit theoretically, the algorithms that will run on them [21]. One of these algorithms, possibly the most famous of them all, is Shor's algorithm [2]. Shor's algorithm shows that the factorization problem of very large numbers, for example those used in RSA keys, could be done in a time in the order of seconds. Currently, to factor a very large number, with computers based on the Von Neumann architecture, it would be necessary to wait a time in the order of years. That is why, nowadays, to secure the connections between devices through the Internet, RSA key pairs are used. With the implementation and commercialization of the first quantum computers, computer security as it is currently implemented, would become obsolete.

In the field of the machine learning, it has designed several algorithms too, like running subroutines of computationally expensive and the translation of stochastic methods into the language of quantum theory [39]. In [33] it has been proposed a Quantum Support Vector Machine(Q-SVM) for classification for Big Data. There it is detailed how a Q-SVM could be implemented in $O(\log NM)$ completion time, for both training and classification stages.

Other work assessing the capacity of Q-SVMs is [16]. It shows that when an SVM is used for a dataset with many features, the computation done by the kernel of the algorithm is very expensive. To solve that problem, it is proposed to use two methods of Q-SVMs that takes advantage of the exponentially large state space that characterizes quantum computing, through controllable entanglement and interference features. A similar approach is proposed in [37].

3.1 Quantum Computing: Background

In order for the design of algorithms within the field of quantum computing to improve performance with respect to their counterparts in classical Von Neumann's computing, quantum mechanical effects must be used. These effects are those that govern the subatomic world as the superposition of states, which says that a subatomic particle, until the time of its measurement, is in a multitude of states intermediate to the two typical states of operation in classical computing (0.1). Thanks to this characteristic it is possible to carry out any operation, the system allows to evaluate all the possibilities in only one step, that is to say, it carries out a parallel computation in a natural way; whereas classically, this evaluation process is carried out in sequential steps [2].

In classical computing, the smallest unit on which to store information is called a bit. A single bit can only store one value between two possible values: 0 or 1. Similarly, in quantum computing, the elementary unit is a qubit. The state of a qubit is determined by the composition of the probabilities it has of being, at the time of measurement its value, in one of its two possible states, $|0\rangle$ and $|1\rangle$.

The states in the q-bit are represented by vectors, belonging to a Hilbert's space L^2 [28]. Being its final value, at the time of measurement, determined by the sum of the probabilities it has of being in each of its possible states $|\psi\rangle = a_0|0\rangle + a_1|1\rangle$, where a_0 and a_1 are complex numbers that follow the normalization relation: $|a_0|^2 + |a_1|^2 = 1$. This means that a given qubit will have, when it is looked, a 0 value with a probability of $|a_0|^2$ and a value of 1 with a probability of $|a_1|^2$.

To work with qubits, it is necessary to use structures where they can be stored, the quantum registers. To represent the information stored in any quantum register, it is used the tensor product of the qubits that are stored in them [41]: $|\phi\rangle = |q_0\rangle \oplus |q_1\rangle$, where $|q_0\rangle$ and $|q_1\rangle$ are the qubits comprised in the quantum registry $\phi\rangle$ and the $\oplus$ is the tensor product symbol.

The state of the record, where it is shown the 4 basic states in which the 2 qubit register can be found, it is defined with $|\phi\rangle = a_0|00\rangle + a_1|01\rangle + a_3|10\rangle + a_4|11\rangle$, where the probabilities follow the rule $1 = |a_0|^2 + |a_1|^2 + \cdots + |a_n|^2$.

In quantum computing, as in classical computing, it is necessary to design and implement circuits that allow the execution of an algorithm. In this case, quantum logic gates are used for the application of logic operations on sets of qubits. The main difference between classical and quantum logic gates is that classical logic gates work on a finite set of values, while quantum logic gates work on continuous data of the Hilbert space [45].

Although, due to the nature of qubits, it is thought that there can be infinite quantum logic gates, it has been demonstrated that by using only two universal logic gates, it is possible to perform any unitary transformation on a set of N qubits [10].

Thanks to the parallelism obtained from the application of the quantum physics theory, it is possible to reduce the execution time of the algorithms that are based on the classical computing based on Von Neumann's architecture, sequential by nature, to an order of logarithmic execution time in either both, supervised and non-supervised machine learning [20].

4 Quantum Algorithm for the K-Nearest Neighbors Classification

The importance of the k-NN algorithm lies in the fact that it is very often used as a subroutine in many machine learning models. For this reason, and as a search for its optimization, quantum versions of this algorithm have been designed, with a theoretic execution time on the order of $O(N)$ [35]. On the other hand, the

k-NN algorithm has a time complexity of $O(NxM)$ [32], where N is the training samples and M the number of dimensions in the dataset.

The basic operation of this algorithm is to measure distances between points in a data set. In this way, when it is intended to classify a sample, the distance between the sample to be classified and the rest of the training dataset is calculated. Examples of, simple and easy to implement, distance measurements algorithms are Euclidean, Manhattan and Hamming distance.

In order to tackle the problem of the computational cost of the k-NN execution, some authors have proposed a quantum version of the algorithm. Making use of the state superposition properties of quantum particles provides a great parallelism in its execution. This is due to the fact that each element of a quantum superposition state is operated simultaneously.

It is possible to demonstrate, although in a theoretical way, how making use of the quantum properties and mechanisms of the state superposition of the quantum particles, we are able to reduce the execution time of an algorithm like k-NN. While in traditional computing it is only possible to store, in a n bit register, a number of the set $\{0, 1, \ldots, i, 2^n - 1\}$, in a n qubits register, it is possible to store that whole binary set of numbers with their corresponding probability $|c_i|^2$.

Thanks to the parallelism obtained by the properties offered by the quantum mechanics of theoretical quantum computers, the speed of execution of the quantum version of the algorithm can be improved. To design this adaptation, we can follow the next steps [35]:

- Prepare the dataset. Data characteristics must be converted to bit vectors, which are mapped for use in quantum computing. The data set can then be represented by N feature vectors $|v^p\rangle$, with $P = 1, 2, \ldots, N$ and whose corresponding class is $c^p \in \{1, 2, \ldots, l\}$, being able to be represented as $\{v_1^p \ldots v_n^p, c^p\} \in \mathbb{H}_2^{\oplus n} \oplus \mathbb{H}_l$. Equation 1 shows the representation of the training set superposition.

$$|\tau\rangle = \frac{1}{\sqrt{N}} \sum_p \{v_1^p \ldots v_n^p, c^p\} \tag{1}$$

- Take a sample and normalize it as a n-dimensional feature vector: $x_1 \ldots x_n$. put in a register the unclassified quantum state and the τ training set in another register, with an ancillary qubit $|0\rangle$ in a third one. In Eq. 2 is shown the result of this step:

$$|\phi_0\rangle = \frac{1}{\sqrt{N}} \sum_p \{x_1 \ldots x_n; v_1^p \ldots v_n^p, c^p; 0\} \tag{2}$$

- Calculate the distance between the sample to be classified $\{x_1 \ldots x_n\}$ and the training set $\{v_1^p \ldots v_n^p, c^p\}$. Any distance metric can be used for this, e.g. in [35] it is used the Hamming distance metric. The result of the metric is then stored in the first register as $\{d_1^p \ldots d_n^p\}$. Equation 3 shows the state of the registers.

$$|\phi_1\rangle = \frac{1}{\sqrt{N}} \sum_p \{d_1 \ldots d_n; v_1^p \ldots v_n^p, c^p; 0\} \tag{3}$$

– The training set of data, that are closer to the chosen sample than a t threshold value, are labeled in this step. The k data that is closer than the threshold is changed to the auxiliary qubit $|0\rangle$ to $|1\rangle$. In Eq. 4 it is formally defined this step:

$$|\phi_2\rangle = \frac{1}{\sqrt{N}}\left(\sum_{p\in\Omega}|d_1\dots d_n; v_1^p\dots v_n^p, c^p; 1\rangle + \sum_{p\notin\Omega}|d_1\dots d_n; v_1^p\dots v_n^p, c^p; 0\rangle\right) \tag{4}$$

where Ω determines the indices of the training data, whose distance to the sample is less than the threshold t.

– Finally, the labels with a $|1|ranlge$ in the auxiliary qubit are selected from the set of samples by means of a projection operator $\Gamma = \nVdash \oplus |1\rangle\langle 1|$. Equation 5 shows how the samples have been selected, having only the vectors $|v^p\rangle$ that are k closest to the sample to classify:

$$|\phi_3\rangle = \Gamma|\phi_2\rangle = \alpha\sum_{p\in\Omega}|d_1\dots d_n; v_1^p\dots v_n^p, c^p; 1\rangle \qquad st. \sum|\alpha|^2 = 1 \tag{5}$$

being α the renormalized amplitude of each component in $|\phi_3\rangle$. Measuring c^p, we are able to determine now the category the sample x^p belongs to.

5 Conclusions and Future Work

In this work, we have made an introduction to the quantum mechanics and how they can be applied in a theoretical quantum computer. To make more concrete the study of this work, we have made a thorough study of machine learning algorithms, along with their classification, which will complement the study of the implementations of machine learning algorithms with quantum physics.

It has been shown some studies that have designed a quantum algorithm in the field of machine learning, obtaining in all of them, a great improvement in the theoretical execution time of their classical counterparts. In addition, it has been done a study of some of the flaws that have to be overcome in order to finally implement a quantum computer.

Finally, we have studied, in a minimalistic way, how it is designed the QKNN algorithm. Thanks to this study, the foundations have been laid on studies on other quantum algorithms and in the design of new ones. Also, it will help people that are starting to study this field, by having a good study of the art in this work.

This work will be continued in another study in which the implications of the quantum era will be shown, in the blockchain field [11]. The use of blockchain technology is one of the hottest topics in the world, since it brings an optimization and improvement to most of the current IoT systems [23–25]. In addition, and due to the high dependence that this technology has on the use of cryptographic keys, for the user's identification [26,44], this technology will be seriously affected if Shor's algorithm is implemented in a commercialized quantum computer.

Acknowledgements. This work has been partially supported by the European Regional Development Fund (ERDF) through the Interreg Spain-Portugal V-A Program (POCTEP) under grant 0677_DISRUPTIVE_2_E (Intensifying the activity of Digital Innovation Hubs within the PocTep region to boost the development of disruptive and last generation ICTs through cross-border cooperation). Also, the research of Yeray Mezquita is supported by a pre-doctoral fellowship granted by the University of Salamanca and cofinanced by Banco Santander.

References

1. Alonso, R.S., Tapia, D.I., Bajo, J., García, Ó., de Paz, J.F., Corchado, J.M.: Implementing a hardware-embedded reactive agents platform based on a service-oriented architecture over heterogeneous wireless sensor networks. Ad Hoc Netw. **11**(1), 151–166 (2013)
2. Caicedo-Ortiz, H.E.: Algoritmo de factorización para un computador cuántico. Latin-Am. J. Phys. Educ. **4**(2), 13 (2010)
3. Candanedo, I.S., Nieves, E.H., González, S.R., Martín, M.T.S., Briones, A.G.: Machine learning predictive model for industry 4.0. In: International Conference on Knowledge Management in Organizations, pp. 501–510. Springer, Heidelberg (2018)
4. Castellanos-Garzón, J.A., Costa, E., Corchado, J.M., et al.: An evolutionary framework for machine learning applied to medical data. Knowl.-Based Syst. **185**, 104982 (2019)
5. Chamoso, P., Rivas, A., Rodríguez, S., Bajo, J.: Relationship recommender system in a business and employment-oriented social network. Inf. Sci. **433**, 204–220 (2018)
6. De Paz, J.F., Tapia, D.I., Alonso, R.S., Pinzón, C.I., Bajo, J., Corchado, J.M.: Mitigation of the ground reflection effect in real-time locating systems based on wireless sensor networks by using artificial neural networks. Knowl. Inf. Syst. **34**(1), 193–217 (2013)
7. Einstein, A.: Über einen die Erzeugung und Verwandlung des Lichtes betreffenden heuristischen Gesichtspunkt. Annalen der Physik **322**(6), 132–148 (1905)
8. Faia, R., Pinto, T., Vale, Z., Corchado, J.M.: Case-based reasoning using expert systems to determine electricity reduction in residential buildings. In: 2018 IEEE Power & Energy Society General Meeting (PESGM), pp. 1–5. IEEE (2018)
9. Feynman, R.P.: Simulating physics with computers. Int. J. Theoret. Phys. **21**(6), 467–488 (1982)
10. Galindo, A., Martin-Delgado, M.A.: Information and computation: classical and quantum aspects. Rev. Modern Phys. **74**(2), 347 (2002)
11. Gao, Y.L., Chen, X.B., Chen, Y.L., Sun, Y., Niu, X.X., Yang, Y.X.: A secure cryptocurrency scheme based on post-quantum blockchain. IEEE Access **6**, 27205–27213 (2018)
12. García, Ó., Prieto, J., Alonso, R., Corchado, J.: A framework to improve energy efficient behaviour at home through activity and context monitoring. Sensors **17**(8), 1749 (2017)
13. González-Briones, A., Hernández, G., Corchado, J.M., Omatu, S., Mohamad, M.S.: Machine learning models for electricity consumption forecasting: a review. In: 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS), pp. 1–6. IEEE (2019)

14. González-Briones, A., Rivas, A., Chamoso, P., Casado-Vara, R., Corchado, J.M.: Case-based reasoning and agent based job offer recommender system. In: The 13th International Conference on Soft Computing Models in Industrial and Environmental Applications, pp. 21–33. Springer, Heidelberg (2018)
15. Gruber, M.: Improving Efficiency by Shrinkage: The James–Stein and Ridge Regression Estimators. Routledge (2017)
16. Havlíček, V., Córcoles, A.D., Temme, K., Harrow, A.W., Kandala, A., Chow, J.M., Gambetta, J.M.: Supervised learning with quantum-enhanced feature spaces. Nature **567**(7747), 209–212 (2019)
17. Heisenberg, W.: The Physical Principles of the Quantum Theory. University of Chicago Press (1930)
18. Kanj, S., Abdallah, F., Denoeux, T., Tout, K.: Editing training data for multi-label classification with the k-nearest neighbor rule. Pattern Anal. Appl. **19**(1), 145–161 (2016)
19. Liu, Z.G., Pan, Q., Dezert, J., Martin, A.: Adaptive imputation of missing values for incomplete pattern classification. Pattern Recogn. **52**, 85–95 (2016)
20. Lloyd, S., Mohseni, M., Rebentrost, P.: Quantum algorithms for supervised and unsupervised machine learning. arXiv preprint arXiv:1307.0411 (2013)
21. Loceff, M.: A course in quantum computing, vol. 1 (2015)
22. Matsunaga, Y., Matsui, K., Nakatoh, Y., Kato, Y.O., Lopez-Sanchez, D., Rodriguez, S., Corchado, J.M.: Preliminary study of mobile device-based speech enhancement system using lip-reading. In: International Symposium on Distributed Computing and Artificial Intelligence, pp. 308–315. Springer, Heidelberg (2018)
23. Mezquita, Y., Casado, R., Gonzalez-Briones, A., Prieto, J., Corchado, J.M.: Blockchain technology in IoT systems: review of the challenges. Ann. Emerg. Technol. Comput. (AETiC) **3**(5), 17–24 (2019)
24. Mezquita, Y., Gazafroudi, A.S., Corchado, J., Shafie-Khah, M., Laaksonen, H., Kamišalić, A.: Multi-agent architecture for peer-to-peer electricity trading based on blockchain technology. In: 2019 XXVII International Conference on Information, Communication and Automation Technologies (ICAT), pp. 1–6. IEEE (2019)
25. Mezquita, Y., González-Briones, A., Casado-Vara, R., Chamoso, P., Prieto, J., Corchado, J.M.: Blockchain-based architecture: a MAS proposal for efficient agri-food supply chains. In: International Symposium on Ambient Intelligence, pp. 89–96. Springer, Heidelberg (2019)
26. Mezquita, Y., Valdeolmillos, D., González-Briones, A., Prieto, J., Corchado, J.M.: Legal aspects and emerging risks in the use of smart contracts based on blockchain. In: International Conference on Knowledge Management in Organizations, pp. 525–535. Springer, Heidelberg (2019)
27. Namazkhan, M., Albers, C., Steg, L.: A decision tree method for explaining household gas consumption: the role of building characteristics, socio-demographic variables, psychological factors and household behaviour. Renew. Sustain. Energy Rev. **119**, 109542 (2020)
28. Nielsen, M.A., Chuang, I.: Quantum computation and quantum information (2002)
29. Planck, M.: Über das Gesetz der Energieverteilung im Normalspectrum. Annalen der Physik **309**(3), 553–563 (1901)
30. De la Prieta, F., Rodríguez-González, S., Chamoso, P., Corchado, J.M., Bajo, J.: Survey of agent-based cloud computing applications. Future Gen. Comput. Syst. **100**, 223–236 (2019)

31. Querejeta, M.U., Alonso, R.S.: Modeling air quality and cancer incidences in proximity to hazardous waste and incineration treatment areas. In: Chavarriaga, J., Diaz, C., Florez, H., García, O., Guarda, T. (eds.) Second International Workshop on Data Engineering and Analytics (WDEA 2019), pp. 108–122. No. 2486 in CEUR Workshop Proceedings, Aachen (2019). http://ceur-ws.org/Vol-2486/#icaiw_wdea_6
32. Raschka, S.: Stat 479: Machine learning lecture notes (2018)
33. Rebentrost, P., Mohseni, M., Lloyd, S.: Quantum support vector machine for big data classification. Phys. Rev. Lett. **113**(13), 130503 (2014)
34. Rivas, A., Chamoso, P., González-Briones, A., Corchado, J.: Detection of cattle using drones and convolutional neural networks. Sensors **18**(7), 2048 (2018)
35. Ruan, Y., Xue, X., Liu, H., Tan, J., Li, X.: Quantum algorithm for k-nearest neighbors classification based on the metric of hamming distance. Int. J. Theoret. Phys. **56**(11), 3496–3507 (2017)
36. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. Pearson Education Limited, Malaysia (2016)
37. Schuld, M., Killoran, N.: Quantum machine learning in feature Hilbert spaces. Phys. Rev. Lett. **122**(4), 040504 (2019)
38. Schuld, M., Petruccione, F.: Machine Learning, pp. 21–73. Springer, Cham (2018)
39. Schuld, M., Sinayskiy, I., Petruccione, F.: An introduction to quantum machine learning. Contemp. Phys. **56**(2), 172–185 (2015)
40. Sen, P.C., Hajra, M., Ghosh, M.: Supervised classification algorithms in machine learning: a survey and review. In: Emerging Technology in Modelling and Graphics, pp. 99–111. Springer, Heidelberg (2020)
41. Shang, S., Shi, M., Shang, W., Hong, Z.: A text classification algorithm based on quantum information. In: 2015 11th International Conference on Natural Computation (ICNC), pp. 381–384. IEEE (2015)
42. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. **26**(5), 1484–1509 (1997)
43. Tapia, D.I., Fraile, J.A., Rodríguez, S., Alonso, R.S., Corchado, J.M.: Integrating hardware agents into an enhanced multi-agent architecture for ambient intelligence systems. Inf. Sci. **222**, 47–65 (2013)
44. Valdeolmillos, D., Mezquita, Y., González-Briones, A., Prieto, J., Corchado, J.M.: Blockchain technology: a review of the current challenges of cryptocurrency. In: International Congress on Blockchain and Applications, pp. 153–160. Springer, Heidelberg (2019)
45. Yang, N.: KNN algorithm simulation based on quantum information (2019)