Scientific Research Publishing

Smart Grid and Renewable Energy, 2016, 7, 293-301
http://www.scirp.org/journal/sgre
ISSN Online: 2151-4844
ISSN Print: 2151-481X

Prediction of Solar Irradiation Using Quantum Support Vector Machine Learning Algorithm

Makhamisa Senekane[1], Benedict Molibeli Taele[2]

[1]Faculty of Computing, Botho University-Maseru Campus, Maseru, Lesotho
[2]Department of Physics and Electronics, National University of Lesotho, Roma, Lesotho
Email: makhamisa.senekane@bothouniversity.ac.bw, bm.taele@nul.ls

How to cite this paper: Senekane, M. and Taele, B.M. (2016) Prediction of Solar Irradiation Using Quantum Support Vector Machine Learning Algorithm. *Smart Grid and Renewable Energy*, 7, 293-301.
http://dx.doi.org/10.4236/sgre.2016.712022

Received: October 6, 2016
Accepted: December 3, 2016
Published: December 6, 2016

Abstract

Classical machine learning, which is at the intersection of artificial intelligence and statistics, investigates and formulates algorithms which can be used to discover patterns in the given data and also make some forecasts based on the given data. Classical machine learning has its quantum part, which is known as quantum machine learning (QML). QML, which is a field of quantum computing, uses some of the quantum mechanical principles and concepts which include superposition, entanglement and quantum adiabatic theorem to assess the data and make some forecasts based on the data. At the present moment, research in QML has taken two main approaches. The first approach involves implementing the computationally expensive subroutines of classical machine learning algorithms on a quantum computer. The second approach concerns using classical machine learning algorithms on a quantum information, to speed up performance of the algorithms. The work presented in this manuscript proposes a quantum support vector algorithm that can be used to forecast solar irradiation. The novelty of this work is in using quantum mechanical principles for application in machine learning. Python programming language was used to simulate the performance of the proposed algorithm on a classical computer. Simulation results that were obtained show the usefulness of this algorithm for predicting solar irradiation.

Keywords

Quantum, Quantum Machine Learning, Machine Learning, Support Vector Machine, Quantum Support Vector Machine, Energy, Solar Irradiation

1. Introduction

Machine learning is a subfield of artificial intelligence. It is a set of techniques that are used to analyze and find patterns in input data to make predictions/inferences [1]-[10]. It has applications in areas such as image recognition, natural language processing, robotics, spam filtering, drug discovery, medical diagnosis, financial analysis, bioinformatics, marketing and even politics [10] [11] [12].

There are various classical machine learning algorithms, and these include Bayesian networks, artificial neural networks, deep learning, clustering and Support Vector Machine (SVM) to name but a few. The main focus of this paper is on the quantum version of SVM algorithm, which was introduced by Vapnik in the 1990s [13]. Machine learning algorithms can be divided into three major categories, namely supervised learning, unsupervised learning and reinforcement learning, depending on the type of data to be used for predictive analytics [1] [3] [10] [13].

The field of Quantum Information Processing (QIP) exploits quantum mechanical concepts such as superposition, entanglement and tunneling for computation and communication tasks [14]. Recently, there has been a concerted effort to explore the benefits of using QIP for machine learning applications. This results in the field of Quantum Machine Learning (QML). It has also been demonstrated that QML techniques provide a performance speedup compared to their classical counterparts [11] [15]. This speedup is the major motivation for exploring QML algorithms.

There are two basic approaches to QML [9]. The first approach uses the classical data as input, and transforms it into quantum data so that it could be processed on a quantum computer. In essence, this approach implements classical machine learning algorithms on a quantum computer. The second approach involves making use of quantum mechanical principles in order to design machine learning algorithms for classical computers. In the work reported in this paper, we used the first approach to model solar power using quantum SVM.

The remainder of this paper is structured as follows. The next section provides background information on machine learning, QIP and QML. This is followed by Section 3, which discusses the design and implementation of the sun power prediction model reported in this Manuscript. Section 4 provides the results and discusses the results obtained. Finally, the last section concludes this paper.

2. Background Information

Machine learning, which is used interchangeably with predictive analytics, is a sub-field of artificial intelligence which is concerned with building algorithms that make use of input data to make predictions [1] [2] [3] [4]. There are three main categories of machine learning, and they are [1] [10]:

-Supervised learning: makes use of both training data and data label to make predictions about future points. Examples of supervised learning algorithms are logistic regression, artificial neural networks and support vector machines.

-Unsupervised learning: makes use of training data only to make a model that maps inputs to output. As opposed to supervised learning, unsupervised learning does not make use of data label. Examples of unsupervised learning are clustering and anomaly detection algorithms.

-Reinforcement learning: uses reinforcement in the form of reward or punishment. If the algorithm succeeds in making correct predictions, it is rewarded. However, if it fails, it is punished. Reinforcement learning is used mainly in robotics and computer games.

2.1. Support Vector Machines

Support vector machine learning is the most commonly used "off-the-shelf" supervised learning algorithm [1]. SVM solves problems in both classification and regression. It uses the principle of maximum margin classifier to separate data. For a d-dimensional data, SVM uses a d – 1 hyperplane for data separation. For instance, if data are supplied on a plane (two dimensions), SVM would use a line (one dimension) for classification. The principle of maximum margin classification ensures that there is a maximum separation between positive results ($y = 1$) and negative results ($y = -1$). The margin in this case is the distance between the decision boundary and the support vectors, where support vectors are data points closest to the decision boundary.

One of the key advantages of support vector machines is that unlike other supervised learning algorithms, its loss function is a global optimization problem, hence it is not prone to local optima [4]. Additionally, SVM is robust against over-fitting, hence it is suitable for making generalizations even with a small dataset. Lastly, by using a technique known as kernel trick, SVM can separate data which is not linearly separable in its input space. This technique enables SVM to transform input data into higher-dimensional space, where a separating linear hyperplane can be found.

2.2. Quantum Information Processing

In stark contrast to classical computers, which use a binary digit (bit) as a unit of information, quantum computers use a quantum bit (qubit) as a unit of information. Mathematically, a qubit is given as [14] [16]

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

where α and β are probability amplitudes. These amplitudes satisfy the condition

$$|\alpha|^2 + |\beta|^2 = 1 \tag{2}$$

It is worth noting that a qubit, which is a unit of information for a two-state system, can be generalized to any arbitrary d-state. Such a generalized unit of information is known as a quantum digit (qudit) [16]. Just like a classical computer, which use gates for computation, quantum computers also use quantum gates to perform operations on qudits. Essentially, a quantum gate operation on a quantum state $|\psi\rangle$ (which is represented as a column vector) is a linear operation. Therefore, mathematically speaking, quantum information processing makes use of vectors, matrices and tensors, hence it involves linear transformations.

2.3. Quantum Machine Learning

Machine learning generally represents data in vector and matrix form. This is also the case with QIP, hence why QIP concepts find applications in machine learning. This results in the new field of research called quantum machine learning. Quantum machine learning can take two forms: where classical machine learning algorithms are transformed into their quantum counterparts; to be implemented on a quantum information processor, or taking some of the computationally expensive classical machine learning sub-routines and implementing them on the quantum computer.

2.4. Model Evaluation and Validation

Different measures are used to evaluate and validate models. These measures include mean squared error (MSE), Root mean squared error (RMSE), mean absolute error (MAE), and R^2 error.

2.4.1. Mean Squared Error

Mean squared error is one of the measures of the goodness of fit. It measures the closeness of a data line to the data points. For n as the number of predictions, $\tilde{y}$ as the vector of predicted values, and Y as the vector of observations, MSE is given as

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(\tilde{y}_i - Y_i\right)^2 \tag{3}$$

2.4.2. Root Mean Squared Error

Root mean squared error, which is also a measure of goodness of fit, is the average Euclidean distance of the line from the data points. It is given as

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\tilde{y}_i - Y_i\right)^2} \tag{4}$$

where n is the number of predictions, $\tilde{y}$ is the vector of predicted values, and Y is the vector of observations.

2.4.3. Mean Absolute Error

Mean absolute error measures the closeness of predicted results to the observations. It is given as

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|\tilde{y}_i - Y_i\right| \tag{5}$$

2.4.4. R^2 Error

R^2 error is also known as coefficient of determination. It is the measure of degree of variance. It is given as

$$R^2 = s - \frac{\text{SStot}}{\text{SSres}} \tag{6}$$

where, for a mean of observations $\bar{\text{y}}$, SStot is given as

$$\text{SStot} = \sum_{i=1}^{n}\left(Y_i - \bar{\text{y}}_i\right)^2 \tag{7}$$

and

$$\text{SSres} = \sum\nolimits_{i=1}^{n} \left(\tilde{y}_i - Y_i \right)^2 \tag{8}$$

3. Implementation

In this work, quantum support vector machine was implemented using a recorded data from Digital Technology Group (DTG) Weather Station in Cambridge University[1]. The dataset consists of forty nine instances, which are the training examples. These instances represent the measurements that were recorded at DTG, with a time interval of thirty minutes. Additionally, this dataset consists of three features, namely temperature, humidity and wind speed.

The recorded classical information is converted to quantum state such that for a training example $\ddot{x}$ and number of training examples *N*:

$$\left| \ddot{x} \right\rangle = \frac{1}{\left| \ddot{x} \right|} \sum\nolimits_{j=1}^{n} \left(\ddot{x} \right) j \left| j \right\rangle \tag{9}$$

This is then followed by optimizing the quantum support vector hyperplane parameters, as articulated in [17]. The optimization is done by reducing this optimization problem into a system of linear equations, and then using a quantum algorithm for solving a system of linear equations, which uses matrix inversion. This quantum algorithm is known to have an exponential speedup over its classical counterpart.

The quantum support vector machine was implemented using Python programming language.

Python machine learning package used for this task was Scikit-learn version 0.18.0 [5]. The graphical user interface (GUI) part of the implementation was realized using Orange data mining software package, release number 3.3.8[2]. This GUI helped visualize the input dataset and the plots for the results obtained from this implementation. It also supports other python packages such as scikit-learn.

The results were then recorded and errors calculated. The following errors were calculated, for different training sizes:

_mean square error (MSE),
_root mean square error (RMSE),
_mean absolute error (MAE),
_coefficient of determination, R^2.

4. Results and Discussion

The dataset was broken down into different portions, with some part being used for training data, and the other part being used for cross-validation. Table 1 shows different calculated errors for different training data sizes. From the table, it can be observed that the best results are obtained when the training size is 70% of the dataset. Therefore, the training size of 70% was chosen for this implementation.

[1] The official website of DTG is: http://www.cl.cam.ac.uk/research/dtg/weather/

[2] The official website for Orange software package is: http://orange.biolab.si/

The next step was to analyze the correlation of the three features used (temperature, humidity and wind speed). Figure 1 and Figure 2 show the scatter plots of these correlations. Since the graphs in the figures are not linear, it implies that the features were not correlated, hence they were independent. Finally, Sieve diagrams were plotted, and are shown in Figure 3 and Figure 4. These results underline the robustness of the proposed algorithm.

5. Conclusions

We have reported an algorithm for solar power prediction using quantum support vector machine learning algorithm. The algorithm is a quantum counterpart of a classical support vector machine, which is known to have a unique solution, and hence it converges

Table 1. Calculated errors for different dataset training sizes.

Training Size (%)	MSE	RMSE	MAE	R^2
60	3.629	1.905	1.435	0.978
66	3.098	1.760	1.283	0.826
70	2.643	1.626	1.171	0.852
75	2.992	1.730	1.259	0.835
80	3.014	1.736	1.257	0.835

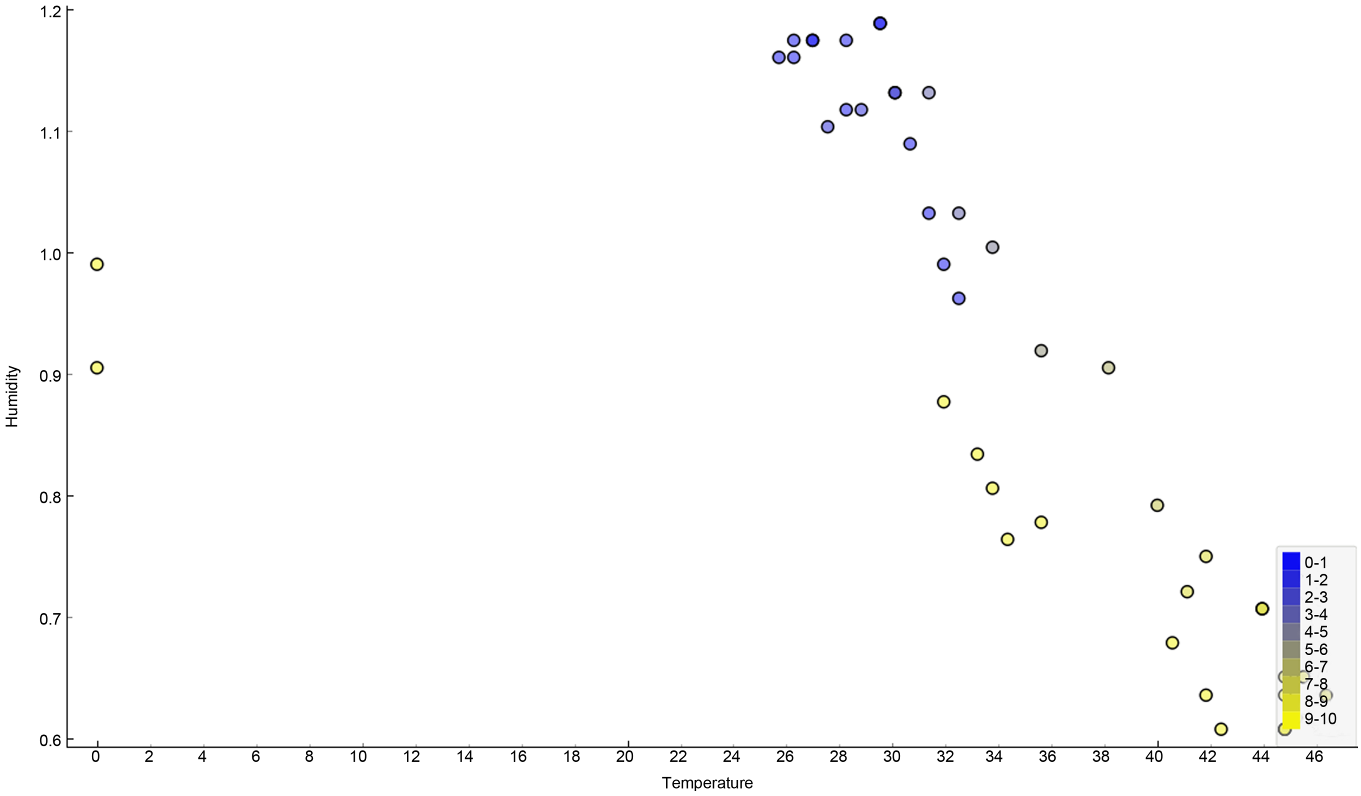

Figure 1. This figure shows the relationship between temperature (in degrees Celsius) and humidity. The non-linearity of the data points implies that the two features are not correlated.

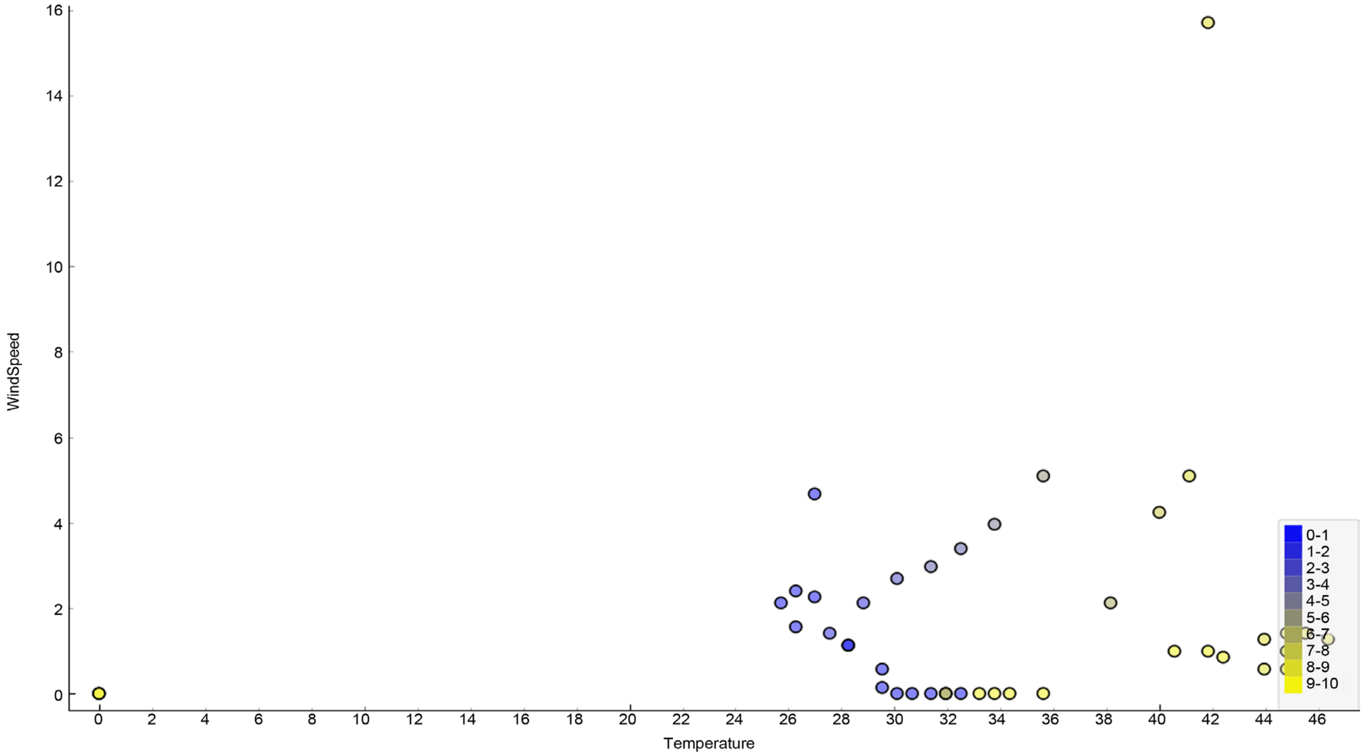

Figure 2. This figure shows the relationship between temperature (in degrees Celsius) and wind speed (in knots). Since the data point portray non-linearity, it can be observed that these two features are independent.

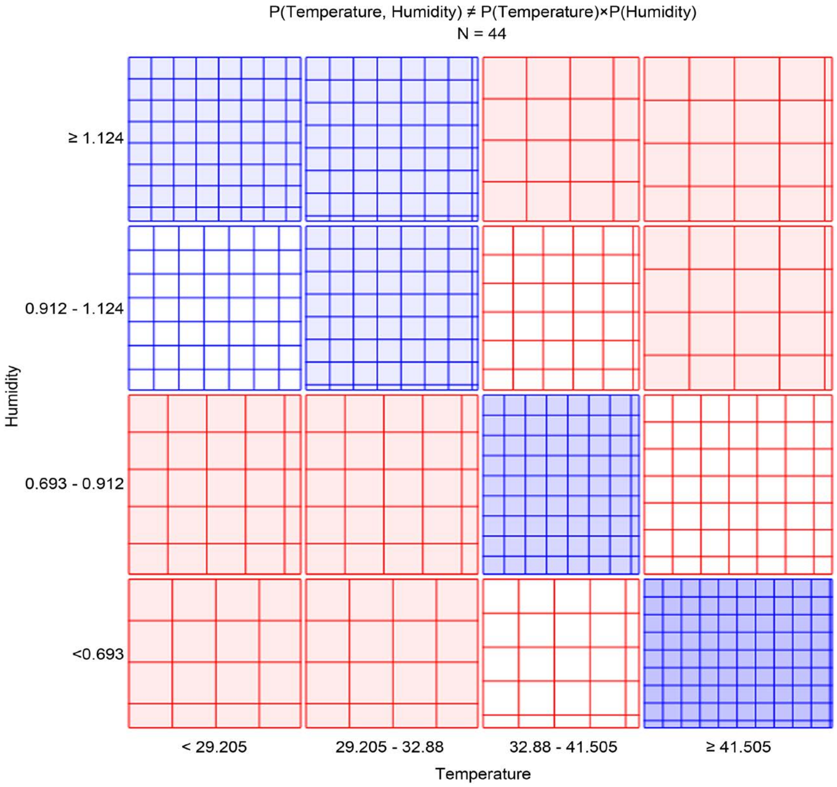

Figure 3. A sieve diagram for temperature and humidity attributes.

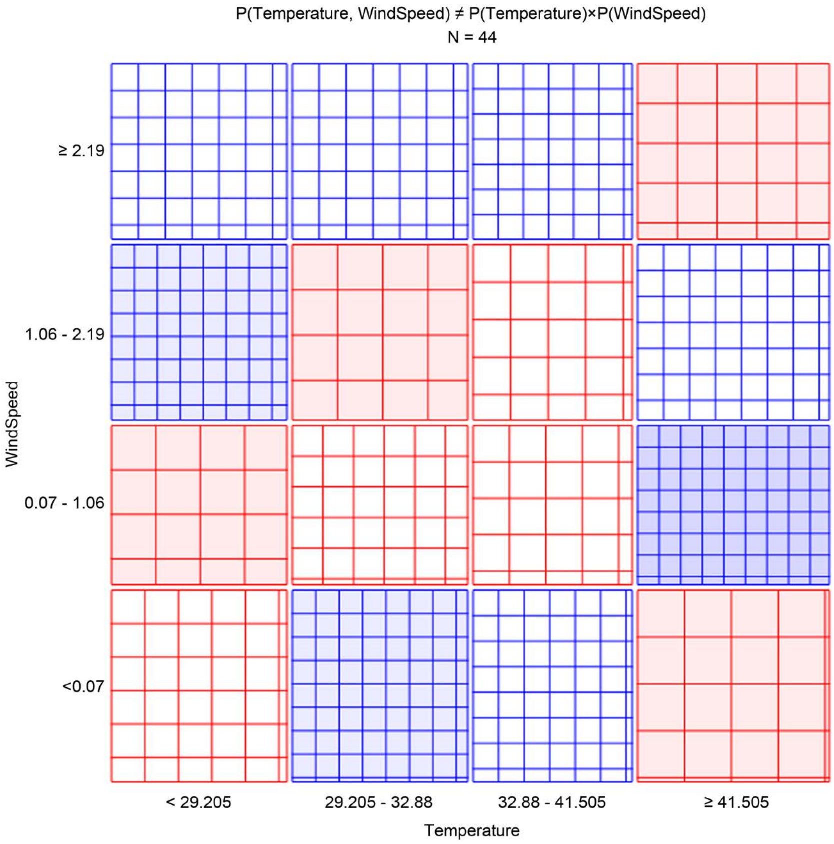

Figure 4. A sieve diagram for temperature and wind speed attributes.

to a global optimum. This is in contrast to other machine learning algorithms such as neural networks, which can converge to local optima, since they may not have unique solutions.

In the work reported in this paper, the quantum support vector algorithm was simulated using Python programming language. A dataset with forty nine instances and three features (temperature, humidity and windspeed) was used for this simulation. The results obtained from the simulation underline the utility of the proposed quantum support vector algorithm for solar power prediction. However, it should be noted that in the implementation, a generic optimization algorithm was used for implementing quantum SVM. Future work should explore the feasibility.

References

[1] Russell, S.J., Norvig, P., Canny, J.F., Malik, J.M. and Edwards, D.D. (2010) Artificial Intelligence: A Modern Approach. Prentice Hall, New York.

[2] Rogers, S. and Girolami, M. (2015) A First Course in Machine Learning. CRC Press, London.

[3] Sugiyama, M. (2015) Introduction to Statistical Machine Learning. Morgan Kaufmann, Amsterdam.

[4] Bishop, C.M., *et al.* (2006) Pattern Recognition and Machine Learning. Springer, New York.

[5] Garreta, R. and Moncecchi, G. (2013) Learning Scikit-Learn: Machine Learning in Python.

Packt Publishing Ltd., Birmingham.

[6] Raschka, S. (2015) Python Machine Learning. Packt Publishing Ltd., Birmingham.

[7] Ivezic, Z., Connolly, A., Vanderplas, J. and Gray, A. (2014) Statistics, Data Mining and Machine Learning in Astronomy. Princeton University Press, Princeton, New Jersey.

[8] Lantz, B. (2013) Machine learning with R. Packt Publishing Ltd., Birmingham.

[9] Wittek, P. (2014) Quantum Machine Learning: What Quantum Computing Means to Data Mining. Academic Press, Cambridge, Massachusetts.

[10] Schuld, M., Sinayskiy, I. and Petruccione, F. (2015) An Introduction to Quantum Machine Learning. *Contemporary Physics*, **56**, 172-185.

[11] Cai, X.D., Wu, D., Su, Z.E., Chen, M.C., Wang, X.L., Li, L., Liu, N.L., Lu, C.Y. and Pan, J.W. (2015) Entanglement-Based Machine Learning on a Quantum Computer. *Physical Review Letters*, **114**, 110504. https://doi.org/10.1103/PhysRevLett.114.110504

[12] Siegel, E. (2013) Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Die. John Wiley & Sons, Hoboken, New Jersey.

[13] Marsland, S. (2015) Machine Learning: An Algorithmic Perspective. CRC Press, Boca Raton, Florida.

[14] Nielsen, M.A. and Chuang, I.L. (2010) Quantum Computation and Quantum Information. Cambridge University Press, Cambridge, UK. https://doi.org/10.1017/CBO9780511976667

[15] Lloyd, S., Mohseni, M. and Rebentrost, P. (2013) Quantum Algorithms for Supervised and Unsupervised Machine Learning. arXiv:1307.0411

[16] Wilde, M.M. (2013) Quantum Information Theory. Cambridge University Press, Cambridge, UK. https://doi.org/10.1017/CBO9781139525343

[17] Li, Z., Liu, X., Xu, N. and Du, J. (2015) Experimental Realization of a Quantum Support Vector Machine. *Physical Review Letters*, **114**, 140504. https://doi.org/10.1103/PhysRevLett.114.140504