Quantum SVR for Chlorophyll Concentration Estimation in Water with Remote Sensing

Edoardo Pasetto, Morris Riedel, *Member, IEEE,* Farid Melgani, *Fellow, IEEE,*
Kristel Michielsen, Gabriele Cavallaro, *Member, IEEE*

Abstract—**The increasing availability of quantum computers motivates researching their potential capabilities in enhancing the performance of data analysis algorithms. Similarly, as in other research communities, also in Remote Sensing (RS) it is not yet defined how its applications can benefit from the usage of quantum computing. This paper proposes a formulation of the Support Vector Regression (SVR) algorithm that can be executed by D-Wave quantum computers. Specifically, the SVR is mapped to a Quadratic Unconstrained Binary Optimization (QUBO) problem that is solved with Quantum Annealing (QA). The algorithm is tested on two different types of computing environments offered by D-Wave: The Advantage system, which directly embeds the problem into the Quantum Processing Unit (QPU), and a Hybrid solver that employs both classical and quantum computing resources. For the evaluation, we considered a biophysical variable estimation problem with RS data. The experimental results show that the proposed quantum SVR implementation can achieve comparable or in some cases better results than the classical implementation. This work is one of the first attempts to provide insight into how QA could be exploited and integrated in future RS workflows based on Machine Learning (ML) algorithms.**

Index Terms—**Support vector regression, quantum computing, quantum annealing, quantum machine learning, remote sensing**

I. INTRODUCTION

REGRESSION analysis is a statistical process whose objective is to find the relationship between a set of independent variables $\{\mathbf{x}\}$ and a dependent variable y [1]. It holds an important role in many applications such as financial forecasting [2], geomagnetic data reconstruction [3], marketing, sociology, epidemiology and risk analysis [4]. In the field of RS, regression analysis has been applied in different applications [5, 6].

In the context of Quantum Machine Learning (QML) [7]–[9] only few works have already addressed regression analysis problems. For instance, a quantum version of a linear

E. Pasetto is with the Jülich Supercomputing Centre, Wilhelm-Johnen Straße 52428 Julich, Germany, and the RWTH Aachen University, D-52056 Aachen, Germany (e-mail: e.pasetto@fz-juelich.de).

Morris Riedel is with the University of Iceland, 107 Reykjavik, Iceland, and the Jülich Supercomputing Centre, Wilhelm-Johnen Straße 52428 Julich, Germany (e-mail: morris@hi.is).

Farid Melgani is with the Department of Information Engineering and Computer Science, University of Trento, Via Sommarive 9, I-38123, Trento, Italy (e-mail: melgani@disi.unitn.it).

Kristel Michielsen is with the Jülich Supercomputing Centre, Wilhelm-Johnen Straße 52428 Jülich, Germany and the RWTH Aachen University, D-52056 Aachen, Germany (e-mail: k.michielsen@fz-juelich.de).

Gabriele Cavallaro is with the Jülich Supercomputing Centre, Wilhelm-Johnen Straße 52428 Jülich, Germany and the University of Iceland, 107 Reykjavik, Iceland (e-mail: g.cavallaro@fz-juelich.de).

regression algorithm and of a ridge regression algorithm have been proposed [10, 11]. Among the different paradigms of Quantum Computing (QC), QA has recently provided promising results in diverse machine learning applications [12, 13]. QA is a metaheuristic for solving combinatorial optimization problems [14, 15] . QA is closely related to Adiabatic Quantum Computing (AQC) [16] which was shown to be polynomially equivalent to the universal gate-based model, which is a different paradigm of QC [17]. However, QA can only solve a specific class of problems and therefore the redefinition of ML algorithms in a suitable format is one of the central design challenges when working with QA-enhanced ML models [18]. This work presents an implementation of the Support Vector Regression (SVR) [19] algorithm that uses QA for solving the optimization problem related to the training phase of the SVR algorithm. Previous works tried to apply QA to optimize the training procedure of a Support Vector Machine for classification tasks [20].

A similar implementation of QA-optimized SVR algorithm was proposed for facial landmarks detection [21]. Specifically, our implementation uses a similar workflow for constructing the QUBO, but the mathematical formulation presents some differences in the constraints enforcement procedure (Section II). Additionally, in this work we propose 6 different methods to combine the solutions returned by the annealer when running the problem on the Advantage system (Section II). Moreover, our implementation was tested on both Hybrid and direct QPU solvers, whereas [21] tested the Quantum Support Vector Regression (QSVR) only on Hybrid solvers. This work focuses on bio-physical parameter estimation related to chlorophyll concentration in water [22]–[24]. The proposed implementation was tested on a synthetic and a real RS dataset related to chlorophyll concentration in water. The quantum system used in the experiments was provided by the company D-Wave. Specifically, the experimental validation was conducted on the D-Wave *Advantage_system4.1* solver and the *hybrid_binary_quadratic_model_version2* hybrid solver. The access to such computational resources was provided through the D-Wave *Leap* cloud service. The purpose of this work is to investigate how QA could improve existing machine learning frameworks for RS applications.

II. QA-BASED IMPLEMENTATION OF SUPPORT VECTOR REGRESSION

A. Support Vector Regression

The mathematical formulation of the ϵ-insentive SVR is now briefly described. Let $T = \{(\mathbf{x}_n, y_n), n = 0, \ldots, N-1\}$ be the

dataset used for the training phase constituted by N training samples. Each of such samples is formed by a feature vector $\mathbf{x}_n \in \mathbb{R}^d$, where d is the dimension of the feature space, and a target value $y_n \in \mathbb{R}$. It can be shown, with some mathematical manipulation, that the training phase amounts to the solving of the following constrained optimization problem:

$$L(\boldsymbol{\alpha}, \hat{\boldsymbol{\alpha}}) = \frac{1}{2} \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} (\alpha_n - \hat{\alpha}_n)(\alpha_m - \hat{\alpha}_m)k(\mathbf{x}_n, \mathbf{x}_m) + \quad (1)$$
$$- \epsilon \sum_{n=0}^{N-1}(\alpha_n + \hat{\alpha}_n) + \sum_{n=0}^{N-1}(\alpha_n - \hat{\alpha}_n)y_n$$

where the constraints to be enforced are:

$$\sum_{n=0}^{N-1}(\alpha_n - \hat{\alpha}_n) = 0 \tag{2a}$$

$$0 \le \alpha_n \le C \tag{2b}$$

$$0 \le \hat{\alpha}_n \le C \tag{2c}$$

In the optimization problem, the terms $\boldsymbol{\alpha} = \{\alpha_n : n = 0, \ldots, N-1\}$ and $\hat{\boldsymbol{\alpha}} = \{\hat{\alpha}_n : n = 0, \ldots, N-1\}$ are the variables with respect to which the problem is optimized. The terms C and ϵ are parameters related to the SVR that controls the overfitting and the error sensitivity, respectively. After finding the values of $\boldsymbol{\alpha}$ and $\hat{\boldsymbol{\alpha}}$ the prediction function is defined as:

$$y(\mathbf{x}) = \sum_{n=1}^{N}(\alpha_n - \hat{\alpha}_n)k(\mathbf{x}, \mathbf{x}_n) + b \tag{3}$$

The value of b can be deducted from any data point for which $0 < \alpha_n < C$ or $0 < \hat{\alpha}_n < C$ using the formula:

$$b = y_n - \epsilon - \sum_{m=1}^{N}(\alpha_m - \hat{\alpha}_m)k(\mathbf{x}_n, \mathbf{x}_m) \tag{4}$$

To obtain a more robust estimation of b it is preferable, however, to average the result from multiple data points [25]

The term $k(\mathbf{x}_n, \mathbf{x}_m)$ indicates the kernel function, in our experiments a Radial Basis Function (RBF) kernel, whose formula is given by: $e^{(-\gamma||\mathbf{x}_n - \mathbf{x}_m||^2)}$, has been employed.

B. QUBO problem formulation

In order to be processed by the Quantum Annealer, a problem must be in the form of either a Ising Spin problem [26] or a QUBO. For our purposes the original optimization problem related to the SVR was turned into a QUBO problem. Such a problem can be expressed according to the following formula:

$$E = \sum_{i \le j} a_i Q_{i,j} a_j \tag{5}$$

Terms $a_i \in \{0, 1\}$ are the binary variables of the QUBO problem and Q is an upper-diagonal matrix called QUBO weight matrix that defines the problem. To turn the original problem into a QUBO it is first necessary to encode the

variables $\boldsymbol{\alpha}$ and $\hat{\boldsymbol{\alpha}}$ in the binary variables a_i. To do so the following encoding strategy is applied [13]:

$$\alpha_n = \sum_{k=0}^{K-1} B^{k-P} a_{Kn+k}, \tag{6}$$

$$\hat{\alpha}_n = \sum_{k=0}^{K-1} B^{k-P} a_{K(N+n)+k}, \tag{7}$$

In the equations 6 and 7 the value K corresponds to the number of logical qubits used to encode each variable, whereas B is the value of the base used for the encoding. From the above equations it is possible to note that the total number of variables of the QUBO problem is $2KN$ and that the first KN variables are used to encode the $\boldsymbol{\alpha}$ variables whereas the last KN ones are used to encode the variables $\hat{\boldsymbol{\alpha}}$. The parameter $P \ge 0$ is used for enabling the usage of negative exponents in the encoding procedure. To enforce the constraint defined in equation 2a a square penalty term whose strength is regulated by the hyperparameter ξ is added to the cost function. The constraints in eq 2b and 2c are implicitly satisfied by the choice of the hyperparameter C: from the equations it is possible to see that the maximum value that each variable can take is:

$$C = \sum_{k=0}^{K-1} B^k. \tag{8}$$

Therefore by choosing a value of C equal or higher than this quantity it is possible to guarantee the enforcing of the constraints. The lower bound is always satisfied because each α_n and $\hat{\alpha}_n$ is non-negative by definition. Moreover, another penalty term, whose influence is controlled by the hyperparameter β, is added to enforce that, for each value of n, at least one of α_n or $\hat{\alpha}_n$ is equal to 0, or equivalently: $\alpha_n \hat{\alpha}_n = 0, n = 0, \ldots, N-1$:

$$\beta \left(\sum_{0}^{N-1} \alpha_n \hat{\alpha}_n \right) \tag{9}$$

By adding the penalty terms to the cost functions and by applying the encoding equations it is possible to obtain the final formulation of the QUBO problem:

$$\sum_{n,m=0}^{N-1} \sum_{i,j=0}^{K-1} \sum_{s,t=0}^{1} a_{K(sN+n)+i} \tag{10}$$
$$\tilde{Q}_{K(sN+n)+i, K(tN+m)+j} a_{K(tN+m)+j}$$

The term $\tilde{Q}$ is a $2KN \times 2KN$ matrix that defines the problem, whose elements are given by:

$$\tilde{Q}_{K(sN+n)+i, K(tN+m)+j} = (-1)^{(1-\delta_{st})} B^{i+j-2P}$$
$$(\frac{1}{2}k(\mathbf{x}_n, \mathbf{x}_m) + \xi - (1-s)t\delta_{nm}\beta) + \tag{11}$$
$$+ \delta_{nm}\delta_{ij}B^{i-P}\delta_{st}(\epsilon + (-1)^{(1-s)(1-t)}t_n)$$

with $n, m \in \{0, \ldots, N-1\}$, $i, j \in \{0, \ldots, K-1\}$ and $s, t \in \{0, 1\}$. Since the QUBO weight matrix Q is upper-triangular, it is obtained from $\tilde{Q}$ by using the formula:

$$Q_{i,j} = \begin{cases} \tilde{Q}_{i,j} + \tilde{Q}_{j,i}, & \text{if } i < j; \\ \tilde{Q}_{i,j}, & \text{if } i = j; \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

This problem formulation is used for the QPU of the Advantage system and the Leap's Hybrid solver, as they both input a binary quadratic problem. In the case of Leap's Hybrid solver the problem is optimized by using both classical and quantum computing resources. The allocation of such resources and the problem decomposition is done automatically by the solver.

The final step for optimizing a problem with the QPU is the *minor embedding* [27]. In this process the QUBO problem is embedded in the hardware architecture used for the annealing. Specifically, each logical variable a_i is mapped to a chain of qubits, i.e., a group of connected qubits used to represent a specific logical variable of the QUBO problem. The reason why chains are needed is that it is not always possible to directly map the optimization problem directly into the hardware topology. The values of the elements $Q_{i,j}$ of the Q matrix, which represent the coefficients associated to the terms $a_i a_j$ of the cost function, are mapped to physical connections of chains of qubits.

C. Solutions combination Techniques

The Advantage system outputs 10000 reads (i.e., solutions) with different levels of energy. In this work, we select the 40 solutions with the lowest energy and we fuse them to compute the final solution (the number 40 is chosen arbitrarily and other options can be considered). To combine them, we propose six approaches based on different weighted average formulas. The predictions of the 40 solutions are then evaluated on the training dataset. To each of them is assigned a score depending on the value of a specific loss function and such scores are then used to obtain the coefficients of the weighted average. Specifically, the scores are calculated by considering the multiplicative inverse of the value of a loss function between the actual and the predicted value, therefore a lower value of the loss will be associated to a higher score value. The loss functions considered are even and non-negative. The combinations methods differ for the choice of the loss function and how the scores are used to get the coefficients. Every method ensures that each weight coefficient is non-negative, its value is lower or equal than 1 and that their sum is equal to 1. A brief description of the used methods is now provided:

- **QSVR 1**: employs a Mean Squared Error (MSE) loss function and the weights coefficients are obtained by dividing each scores by the sum of all of them
- **QSVR 2**: uses a MSE loss function and the coefficients are obtained by applying a softmax operation on the scores
- **QSVR 3**: uses a log-cosh loss function and the coefficients are obtained as in QSVR 1
- **QSVR 4**: uses a log-cosh loss function and the final weights are obtained through the application of softmax on the scores

- **QSVR 5**: Only the best solution in terms of MSE is considered, this is done by setting the weights associated to the best solution to 1 and all the others to 0
- **QSVR 6**: To each solution is assigned the same weight, therefore a simple average is performed

III. DATASET DESCRIPTION

In this section a brief overview and description of the dataset used for the experimental analysis is provided.

- **MERIS**: The first dataset used is a synthetic dataset whose aim is to simulate the concentration of chlorophyll concentration and its relation to optical measurements. The wavelengths considered are the first eight spectral bands of the MERIS sensor (412.5 nm, 442.5 nm, 490 nm, 510 nm, 560 nm, 620 nm, 665 nm, 681.25 nm). The procedure employed to generate the dataset is the one described in [28].
- **SeaBAM**:The dataset contains information about 919 measurements regarding chlorophyll-a water concentration performed in Europe and US. The value of concentration ranges between 0.019 and 32.787 mg/m^3. The sensor used for the measurements is the Sea-Viewing Wide Field of View Sensor (SeaWIFS) and the wavelengths considered in the experiments were 412, 443, 490, 510 and 555 nm.

In both case the feature vector is constructed by considering the spectral measures at different wavelengths whereas the target value is the corresponding chlorophyll concentration.

IV. EXPERIMENTAL ANALYSIS

For each dataset two experiments were conducted: one using the D-Wave Advantage QPU and the other using Leap's Hybrid Binary Quadratic Model (BQM) solver. The implementation of the classical SVR was done using the Python library sci-kit learn. In each setting 10 test runs were carried out, each one using different datasets for training and testing that were randomly chosen from the initial dataset. In each problem instance the results were compared with a traditional SVR on the same datasets. Moreover, the hyperparameters for the quantum and the classical implementation of the SVR were the same for each test run. For the experiments using the Advantage solver the number of training samples was 30, whereas for the hybrid solver the number was 50. The reason for this is that the Advantage system could not always find an embedding with bigger problem instances. This is likely due to the structure of the problem itself that presents many interactions between variables that makes difficult finding an embedding as the number of variables increases [29]. In each test run the hyperparameter γ was validated classically using a validation dataset and a SVR, the validation dataset was divided into 2 parts: the first one was used to train a classical SVR with a given hyperparameter configuration values while the other was used for testing. The configuration that achieved the best performances in terms of mean squared error was then used in the test run. The values of γ were selected from the range $[0.1, 0.5, 1, 1.5, 2, 3, 4, 5, 7, 10, 20, 50]$, whereas the value of C was set to C_{min}, with C_{min} being the quantity

defined in eq. 8, that is equal to the maximum value each α_n and $\hat{\alpha}_n$ can take. The values related to the problem encoding were $B = 4$ and $P = 1$ for the synthetic dataset and $B = 5$ and $P = 0$ for the SeaBAM whereas the number of logical qubits K used to encode the original problem variables was equal to 2 in each test run. For the training phase with the SeaBAM dataset the values of both the feature vector and the target value were converted to the logarithmic domain as was done in [22]–[24]. The reason for this is that the distribution of the bio-physical quantities is assumed to be log-normally distributed [30]. The Table I reports the result in terms of MSE obtained by the hybrid solver on both the synthetic and the seaBaM dataset, while the results obtained by the Advantage system for the synthetic and SeaBAM dataset are reported in Table II and III respectively. When considering the experiments on the synthetic datasets the quantum SVR achieved similar results to its classical counterpart on both the hybrid solver and the Advantage system. In the experiments on the SeaBAM dataset the QSVR on average performed better than the classical one on the hybrid solver whereas in the experiments using the Advantage system the classical implementation performed slightly better but the quantum version managed to obtain good results nevertheless and to perform better on some test runs. The different versions of QSVR performed similarly in the synthetic dataset while on the SeaBAM dataset there was more variation within the results, this might indicate that the correct choice for the solutions combination technique becomes more important as the data complexity increases. In the experiments with the SeaBAM dataset the QSVR 5 managed to obtain the best results in average among the different QSVR implementations but in some specific problem instance it performed worse compared to the other solutions combination techniques. The QSVR 2 obtained the second best average results among the quantum implementations but it was the implementation that outperformed the classical one the most number of times, 5 out of 10. The link to the repository associated to this work can be found at [1]

V. Conclusions

The main objective of this work was to investigate how QA could enhance a SVR algorithm for a RS application. The proposed algorithm was tested on both the D-Wave Advantage system and on the hybrid solver. The results show that the quantum implementation of SVR could achieve similar or in some cases even better results than the classical SVR. This is indicative of the potential of QA especially when considering that the original problem was continuous and unconstrained and that it had to be modified and adapted in order to be solved by the annealer. In general the hybrid solver provided better results than the Advantage system and it could also solve bigger problem instances. Therefore, in the near future practical applications will likely run on a hybrid framework.

Acknowledgment

TABLE I: values of MSE obtained by the classical and quantum SVR implemented on the hybrid solver on both the synthetic and the SeaBAM dataset. Values in bold indicates the results where the quantum implementation performed better.

	MERIS		SeaBAM	
Run	SVR	QSVR	SVR	QSVR
1	0.1035	**0.1032**	6.9309	6.9767
2	0.1011	**0.0882**	8.0448	**2.8367**
3	0.1488	**0.1392**	11.3237	**5.3423**
4	0.0761	0.098	7.9873	**6.0469**
5	0.1134	0.1404	13.7752	**4.9054**
6	0.0973	**0.0955**	5.2729	6.365
7	0.0862	0.0869	3.6089	3.7882
8	0.1117	0.1223	6.4235	8.7816
9	0.1124	0.1225	8.5635	9.9013
10	0.1315	**0.1056**	5.4886	6.019
Average	0.1082	0.1102	7.7419	**6.0963**
Standard deviation	0.0199	0.0187	2.8409	2.0121

References

[1] N. R. Draper and H. Smith, Applied Regression Analysis. John Wiley & Sons, 1998, vol. 326.

[2] T. Trafalis and H. Ince, "Support vector machine for regression and applications to financial forecasting," in Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium, vol. 6, 2000, pp. 348–353 vol.6.

[3] H. Liu, Z. Liu, S. Liu, Y. Liu, J. Bin, F. Shi, and H. Dong, "A nonlinear regression application via machine learning techniques for geomagnetic data reconstruction processing," IEEE Transactions on Geoscience and Remote Sensing, vol. 57, no. 1, pp. 128–140, 2019.

[4] L. Fahrmeir, T. Kneib, S. Lang, and B. Marx, Regression: Models, Methods and Applications. Springer-Verlag Berlin Heidelberg, 9 2013, vol. 9783642343339.

[5] C. H. Whitlock, C. Y. Kuo, and S. R. LeCroy, "Criteria for the use of regression analysis for remote sensing of sediment and pollutants," Remote Sensing of Environment, vol. 12, no. 2, pp. 151–168, 1982. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0034425782900359

[6] Sudaryatno, P. Widayani, T. W. Wibowo, B. A. S. Pramono, Z. N. 'Afifah, A. D. Meikasari, and M. R. Firdaus, "Multiple linear regression analysis of remote sensing data for determining vulnerability factors of landslide in PURWOREJO," IOP Conference Series: Earth and Environmental Science, vol. 500, no. 1, p. 012046, jun 2020. [Online]. Available: https://doi.org/10.1088/1755-1315/500/1/012046

[7] N. Mishra, M. Kapil, H. Rakesh, A. Anand, N. Mishra, A. Warke, S. Sarkar, S. Dutta, S. Gupta, A. Dash, R. Gharat, Y. Chatterjee, S. Roy, S. Raj, V. Jain, S. Bagaria, S. Chaudhary, V. Singh, R. Maji, and P. Panigrahi, Quantum Machine Learning: A Review and Current Status, 01 2021, pp. 101–145.

[8] M. Schuld and F. Petruccione, "Quantum machine learning," in Encyclopedia of Machine Learning and Data Mining. Springer US, 2017, pp. 1034–1043. [Online]. Available: https://doi.org/10.1007/978-1-4899-7687-1_913

[9] A. Sebastianelli, D. A. Zaidenberg, D. Spiller, B. L. Saux, and S. L. Ullo, "On circuit-based hybrid quantum neural networks for remote sensing imagery classification," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 15, pp. 565–580, 2022.

[10] M. Schuld, I. Sinayskiy, and F. Petruccione, "Prediction by linear regression on a quantum computer," Physical Review A, vol. 94, no. 2, Aug 2016.

---

[1] https://gitlab.jsc.fz-juelich.de/sdlrs/quantum-support-vector-regression

TABLE II: Values of MSE obtained on the test set for different combination methods and runs on the synthetic dataset obtained by the Advantage system. Values in bold indicates the results where the quantum implementation performed better.

Run	SVR	QSVR 1	QSVR 2	QSVR 3	QSVR 4	QSVR 5	QSVR 6
1	0.1456	0.1512	0.193	0.1507	0.1504	0.174	0.1518
2	0.1246	0.1546	0.1526	0.1548	0.1561	0.1703	0.1573
3	0.1406	0.1626	0.1528	0.1634	0.1641	0.1594	0.1658
4	0.1276	0.1619	0.1578	0.162	0.1627	0.1647	0.1633
5	0.1405	0.1738	0.1683	0.1744	0.1782	0.1805	0.1841
6	0.112	0.1352	0.1621	0.1356	0.1368	0.1657	0.1382
7	0.143	0.1686	0.1673	0.1674	0.1666	0.1741	0.1665
8	0.1416	0.1566	0.1588	0.1566	0.1563	0.1589	0.1565
9	0.1745	0.1794	**0.1714**	0.1795	0.1786	**0.1725**	0.1818
10	0.1238	0.1369	0.147	0.1368	0.1366	0.147	0.1379
Average	0.1374	0.1581	0.1631	0.1581	0.1586	0.1667	0.1603
Standard deviation	0.0161	0.0137	0.0124	0.0137	0.0139	0.0092	0.0148

TABLE III: Values of MSE obtained on the test set for different combination methods and runs on the SeaBAM dataset obtained by the Advantage system. Values in bold indicates the results where the quantum implementation performed better.

Run	SVR	QSVR 1	QSVR 2	QSVR 3	QSVR 4	QSVR 5	QSVR 6
1	8.0129	**5.2571**	**6.3344**	**5.1476**	**7.3032**	**6.7702**	**7.4246**
2	7.9397	10.127	9.6034	9.8812	9.323	10.3807	9.3034
3	7.0417	7.3176	**6.6154**	7.3818	**6.7388**	7.6413	**6.756**
4	5.2106	8.7789	8.9657	8.9396	8.9745	5.555	8.9751
5	7.1627	**6.5528**	**5.8275**	**5.8984**	**6.0926**	7.2116	**6.148**
6	6.9478	10.6485	9.9555	10.4774	9.676	9.8436	9.6416
7	3.0273	5.688	5.7595	5.7518	5.7307	4.3519	5.7628
8	6.8368	8.0282	**6.1656**	8.6353	8.3695	**5.3047**	8.3542
9	8.4922	9.0675	**7.6096**	8.9369	**7.8412**	**7.4558**	**7.8938**
10	4.6002	5.2455	5.4869	5.497	5.5032	5.5347	5.5033
Average	6.5272	7.6711	7.2324	7.6547	7.5553	7.0049	7.5763
Standard deviation	1.634	1.8767	1.6031	1.8682	1.4419	1.8535	1.4224

[11] C. H. Yu, F. Gao, and Q. Y. Wen, "An Improved Quantum Algorithm for Ridge Regression," IEEE Transactions on Knowledge and Data Engineering, vol. 33, pp. 858–866, 3 2021.

[12] R. Y. Li, R. D. Felice, R. Rohs, and D. A. Lidar, "Quantum annealing versus classical machine learning applied to a simplified computational biology problem," Quantum Information, vol. 4, no. 1, feb 2018. [Online]. Available: https://doi.org/10.1038%2Fs41534-018-0060-8

[13] D. Willsch, M. Willsch, H. De Raedt, and K. Michielsen, "Support Vector Machines on the D-Wave Quantum Annealer," Computer Physics Communications, vol. 248, p. 107006, 2020.

[14] B. Apolloni, C. Carvalho, and D. de Falco, "Quantum stochastic optimization," Stochastic Processes and their Applications, vol. 33, no. 2, pp. 233–244, Dec. 1989. [Online]. Available: https://doi.org/10.1016/0304-4149(89)90040-9

[15] D. de Falco, B. Apolloni, and N. Cesa-Bianchi, "A numerical implementation of quantum annealing," 07 1988.

[16] T. Albash and D. A. Lidar, "Adiabatic quantum computation," Reviews of Modern Physics, vol. 90, no. 1, Jan 2018. [Online]. Available: http://dx.doi.org/10.1103/RevModPhys.90.015002

[17] D. Aharonov, W. van Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev, "Adiabatic quantum computation is equivalent to standard quantum computation," 2005.

[18] W. Guan, G. Perdue, A. Pesah, M. Schuld, K. Terashi, S. Vallecorsa, and J.-R. Vlimant, "Quantum Machine Learning in High Energy Physics," Machine Learning: Science and Technology, vol. 2, p. 011003, 3 2021.

[19] H. Drucker, C. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," in NIPS, 1996.

[20] D. Willsch, M. Willsch, H. De Raedt, and K. Michielsen, "Support vector machines on the d-wave quantum annealer," Computer Physics Communications, vol. 248, p. 107006, Mar 2020. [Online]. Available: http://dx.doi.org/10.1016/j.cpc.2019.107006

[21] A. Dalal, M. Bagherimehrab, and B. Sanders, "Quantum-assisted support vector regression for detecting facial landmarks," 11 2021.

[22] Y. Bazi, N. Alajlan, and F. Melgani, "Improved estimation of water chlorophyll concentration with semisupervised gaussian process regression," IEEE Transactions on Geoscience and Remote Sensing, vol. 50, no. 7, pp. 2733–2743, 2012.

[23] Y. Bazi, N. Alajlan, F. Melgani, H. AlHichri, and R. R. Yager, "Robust estimation of water chlorophyll concentrations with gaussian process regression and iowa aggregation operators," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 7, no. 7, pp. 3019–3028, 2014.

[24] E. Pasolli, F. Melgani, N. Alajlan, and Y. Bazi, "Active learning methods for biophysical parameter estimation," IEEE Transactions on Geoscience and Remote Sensing, vol. 50, no. 10, pp. 4071–4084, 2012.

[25] C. M. Bishop, Pattern Recognition and Machine Learning. Springer, 2006.

[26] F. Barahona, "On the computational complexity of ising spin glass models," Journal of Physics A: Mathematical and General, vol. 15, no. 10, pp. 3241–3253, oct 1982. [Online]. Available: https://doi.org/10.1088/0305-4470/15/10/028

[27] V. Choi, "Minor-embedding in adiabatic quantum computation: I. the parameter setting problem," Quantum Information Processing, vol. 7, no. 5, pp. 193–209, Sep. 2008. [Online]. Available: https://doi.org/10.1007/s11128-008-0082-9

[28] P. Cipollini, G. Corsini, M. Diani, and R. Grasso, "Retrieval of sea water optically active parameters from hyperspectral data by means of generalized radial basis function neural networks," IEEE Transactions on Geoscience and Remote Sensing, vol. 39, no. 7, pp. 1508–1524, 2001.

[29] E. Pelofske, G. Hahn, and H. Djidjev, "Solving large minimum vertex cover problems on a quantum annealer," in Proceedings of the 16th ACM International Conference on Computing Frontiers, ser. CF '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 76–84. [Online]. Available: https://doi.org/10.1145/3310273.3321562

[30] J. W. Campbell, "The lognormal distribution as a model for the bio-optical versatility of the sea," J. Geophys. Res., vol. 100, pp. 13 237–13 254, 1995.