Proceedings of the 7th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/2025.25922

QUBO-Driven Quantum Annealing: A Unified Cloud-Scale Framework for Resource Forecasting, Classification, and Deep Learning Training

Yulian Song[12], Guozheng Zhao[12], Nanxi Wang[3], Qiuxue Zhang[2], Yulin Yang[1], Haoran Gao[4*]

[1]*Department of Financial Management, Chengdu Neusoft University, Chengdu,China*
[2]*Department of Intelligent Systems Application, Chengdu Neusoft University, Chengdu, China*
[3]*College of Management, Sichuan Agricultural University, Chengdu, China*
[4]*Department of Network Engineering, Chengdu Neusoft University, Chengdu, China*
**Corresponding Author. Email: 371553307@qq.com*

Abstract. This paper proposes an integrated QUBO-based quantum-annealing optimization framework for next-generation cloud computing. First, for resource-demand forecasting, we blend Autoregressive (AR) models with Quadratic Unconstrained Binary Optimization (QUBO) and solve the resulting problem via Simulated Annealing; monthly-scale tests reveal superior trend-capture accuracy compared with classical predictors. Second, we embed quantum optimization into Support Vector Machines by casting kernel selection and penalty-parameter tuning as QUBO instances; evaluations on the Iris benchmark—and on synthetically enlarged variants—show marked gains in both classification accuracy and training speed, especially under large-data regimes. Third, we extend the same paradigm to deep learning: a QUBO-formulated quantum-annealing routine optimizes weight initialization and layer-wise learning rates in convolutional networks for image classification, yielding higher top-1 accuracy and reduced wall-clock training time. Across all tasks the hybrid quantum-classical solver consistently outperforms its classical counterparts while maintaining linear scalability in cloud environments. These results show the practical potential of quantum - enhanced algorithms for resource prediction, classification, and deep learning, and pave the way for wider use of quantum technologies in cloud - based AI services.

Keywords: cloud computing, resource demand prediction, support vector machine (SVM), QUBO model, quantum computing, deep learning, simulated annealing algorithm

1. Introduction

Task 1: Forecast monthly cloud resource demand using AR models, convert to QUBO, and solve with Kaiwu SDK’s simulated annealing to predict October requirements [1].

Task 2: Train an SVM classifier on the Iris dataset by reformulating the learning problem as QUBO and optimizing via Kaiwu SDK’s simulated annealing.

DOI: 10.54254/2755-2721/2025.25922

Task 3: Design a CNN for image classification or a recommender model, map its training optimization to QUBO, and accelerate it with Kaiwu SDK's simulated annealing.

2. Problem analysis

Task 1: Fit an AR model to Jan–Sep data, convert it to QUBO, solve with Kaiwu SA, and tune to minimize error for the October forecast.

Task 2:Train an SVM on the 150-sample Iris dataset, cast the margin-maximisation problem into QUBO, and solve it with Kaiwu SDK's simulated annealing for quantum-accelerated binary classification [2].

Task 3: Discretise CNN parameters and loss into a single QUBO objective, solve with Kaiwu SA, then map the binary solution back to continuous weights for quantum-optimised image classification.

3. Assumptions and justifications

We assume the resource-demand data are error-free, the AR model driven by historical data yields unbiased least-square coefficients and white-noise residuals, the chosen lag prevents over-/under-fitting, and parameters remain stable; further, both simulated and quantum annealing solve the QUBO model effectively, with the latter exploiting qubit superposition for parallel search, while all forecasts disregard external shocks or system anomalies [3].

DOI: 10.54254/2755-2721/2025.25922

4. Analysis of global pet industry developments

Table 1. Symbol definition and symbol description

symbol definition	symbol description
y_t	The resource demand value is t
c	constant
φ_i	The coefficients of the regression model
p	Order of regression model
ε_t	Random error variable
a_t	error term
W	weight
b	skewing
ξ_i	slack variable

Data preprocessing:

1. Data preparation: Assume all provided data are authentic and valid.
2. Data cleaning: Check for missing values, fill in with mean or interpolation methods, and remove outliers.
3. Data normalization: Perform normalization if there is significant data fluctuation.

4.1. Model building and solving of task 1

4.1.1. Build an autoregressive (AR) model

To establish an Autoregressive (AR) model, the error sum of squares (ESS) approach is employed to estimate the parameters within the model. This method aims to minimize the discrepancy between predicted values and actual observed data. A target function is constructed using the least squares estimation method to optimize all time points [4].

DOI: 10.54254/2755-2721/2025.25922

$$y_t = c + \sum_{i=1}^{\infty} \varphi_i y_{t-i} + \varepsilon_t$$

4.1.2. The objective function for building a self-regressive model

$$\min_{c,\varphi_1,\varphi_2,...,\varphi_p} \sum_{t=p+1}^{T} \left(\mathbf{y}_t - \mathbf{c} - \sum_{i=1}^{p} \boldsymbol{\varphi}_i y_{t-i} \right)^2$$

4.1.3. Constraint condition

To validate the performance of the QUBO model and adhere to the principle of minimizing errors, additional constraints must be incorporated. Given that the model operates under resource limitations, the prediction results must neither exceed a predefined upper limit nor fall below a specified lower threshold. By implementing rigorous constraint mechanisms, we can ensure the rationality of the solution outcomes. Specifically, we set the minimum threshold at 8,000 and the maximum threshold at 11,000.

4.1.4. The constant term of the discrete parameter and the coefficient of the autoregressive model

$$c = c_{\min} + \delta_c \sum_{k=1}^{b_c} x_{c,k} 2^{k-1}$$

$$\varphi_i = \varphi_{i,\min} + \delta_\varphi \sum_{k=1}^{b_\varphi} x_{\varphi,k} 2^{k-1}$$

4.1.5. Construct the objective function of QUBO model

By substituting the processed discretized parameters into the objective function, we can transform it into a quadratic programming (QUBO) model. This converts the original unconstrained quadratic optimization problem into a quadratic objective function involving binary variables. For time series prediction tasks, the objective function is minimized by fitting the model to reduce the squared error between predicted values and actual values.

$$\text{minimize} \sum_{t=p+1}^{T} \left(\mathrm{y}_t - \mathrm{c} - \sum_{i=1}^{p} \varphi_i y_{t-i} \right)^2$$

DOI: 10.54254/2755-2721/2025.25922

$$E=\sum_{t=p+1}^{T}\left(\mathbf{y}_t-\mathbf{c}-\sum_{i=1}^{p}\boldsymbol{\varphi}_i y_{t-i}\right)^2$$

$$c=c_{\min}+\delta_c\sum_{k=1}^{b_c}x_{c,k}2^{k-1}$$

$$\varphi_i=\varphi_{i,\min}+\delta_\varphi\sum_{k=1}^{b_\varphi}x_{\varphi,k}2^{k-1}$$

$$a_t=\mathbf{y}_t-\left(c_{\min}+\delta_c\sum_{k=1}^{b_c}x_{c,k}2^{k-1}\right)-\sum_{i=1}^{p}\left(\varphi_{i,\min}+\delta_\varphi\sum_{k=1}^{b_\psi}x_{\varphi_t,k}2^{k-1}\right)y_{t-i}$$

$$a_t^2=(a_t)^2$$

$$E=x^TQx+q^Tx+const$$

$$\sum_{t=p+1}^{T}a_t^2=\sum_{t=p+1}^{T}\left[y_t^2-2y_t\left(c+\sum_{i=1}^{p}\varphi_i y_{t-i}\right)+\left(c+\sum_{i=1}^{p}\varphi_i y_{t-i}\right)^2\right]$$

$$\mathbf{minimize} b^TQb$$

$$\min_{x\in\{0,1\}^n}E=\min_{x\in\{0,1\}^n}\left(x^TQx+q^Tx\right)$$

4.2. Model building and solving of task 2

$$\min w^Tw+c\sum_{i=1}^{n}\xi_i$$

DOI: 10.54254/2755-2721/2025.25922

$$\min_{w,b}\frac{1}{2}\parallel w\parallel^2$$

$$y_i\left(\mathrm{w}^{\mathrm{T}}x_i+b\right)\geq 1\quad (\mathrm{i=1,2,...,N})$$

4.2.1. Introduce penalty function and relaxation variable

Since QUBO model can only deal with unconstrained optimization problems, it is necessary to transform the constraints of SVM into the form of penalty function, and introduce relaxation variables and penalty functions to rewrite the constraints:

$$\min_{w,b}\frac{1}{2}\parallel w\parallel^2+c\sum_{i=1}^{n}\xi_i^2$$

The objective is converted into a binary variable to build the QUBO model, so as to minimize the error, and the regularization term is used to avoid overfitting.

4.2.2. Discrete continuous variables

To transform the SVM optimization problem into a QUBO model, we need to convert continuous variables (such as weights and biases) into binary variables. Assuming each weight requires bits of binary representation, the following formula can be used for discretization:

$$w_j=w_{\min}+\sum_{k=1}^{n}b_{jk}\cdot 2^{k-1}\cdot \Delta w$$

4.2.3. Constraint condition

$$\sum_{j=1}^{m}x_y=1,\forall i=1,2,3,4,\ldots,n$$

$$\sum_{i=1}^{n}x_{ij}\leq 1,\forall j=1,2,3,4,\ldots,m$$

$$E\geq C_i,\forall i=1,2,3,4,\ldots,n$$

DOI: 10.54254/2755-2721/2025.25922

$$\text{Minimize} \sum_{i=1}^{n} (E - C_i)^2$$

$$E_{obj} = \sum_{i=1}^{n} \left(E^2 - 2EC_i + C_i^2\right)$$

4.2.4. Build the QUBO objective function

The objective function and constraints are combined into a QUBO model:

$$E_{total} = E_{obj} + \lambda_1 E_{task} + \lambda_2 E_{resource}$$

$$\min \left(\frac{1}{2} \sum_{j=1}^{d} w_j^2 + C \sum_{i=1}^{N} \left[\max\left(0, 1 - y_i \left(w^T x_i + b\right)\right)\right]^2 \right)$$

In this case, each term in the objective function can be expanded into a quadratic form, so as to be suitable for the optimization format of QUBO model.

5. Conclusion

The AR-QUBO hybrid delivers accurate short-term forecasts via binary-discretised autoregression solved by simulated/quantum annealing, but cost, tuning and stochasticity limit scalability; the CNN-QUBO version promises quantum-accelerated global optimisation yet faces qubit noise, quantisation error and matrix-overhead hurdles. We iteratively minimise squared error on discretised parameters and will promote the Q-EAR-CNN framework through industry partnerships and continuous user-driven refinement.

Yulian Song and Guozheng Zhao: Conceptualization, Methodology, Data curation, Writing-Original draft preparation, Visualization, Investigation.

References

[1] Atobe, Y., Tawada, M., & Togawa, N. (2022). Hybrid Annealing Method Based on subQUBO Model Extraction With Multiple Solution Instances. IEEE Transactions on Computers, 71(10), 2618-2631. https: //doi.org/10.1109/TC.2022.3168775.

[2] Glover, F., Kochenberger, G., & Du, Y. (2019). Quantum Bridge Analytics I: A Tutorial on Formulating and Using QUBO Models. Journal of Heuristics. https: //doi.org/10.1007/s10732-019-09419-0.

[3] Wen, J., Wang, Z., Huang, Z., Cai, D., Jia, B., Cao, C., Ma, Y., Wei, H., Wen, K., & Qian, L. (2023). Optical Experimental Solution for the Multi-way Number Partitioning Problem and Its Application to Computing-Power Scheduling. Science China Physics, Mechanics & Astronomy, 66(9), 290313. https: //doi.org/10.1007/s11433-022-1983-8.

[4] Zha, J., Su, J., Li, T., Cao, C., Ma, Y., Wei, H., Huang, Z., Qian, L., Wen, K., & Zhang, J. (2023). Encoding Molecular Docking for Quantum Computers. Journal of Chemical Theory and Computation. https:

DOI: 10.54254/2755-2721/2025.25922

//doi.org/10.1021/acs.jctc.3c00943.