

<https://doi.org/10.1038/s41545-025-00440-y>

# Quantum machine learning regression optimisation for full-scale sewage sludge anaerobic digestion

Yomna Mohamed<sup>1,2,5</sup>, Ahmed Elghadban<sup>1,3,5</sup>, Hei I Lei<sup>1</sup>, Amelie Andrea Shih<sup>1,4</sup> & Po-Heng Lee<sup>1</sup>✉

Anaerobic digestion (AD) is a crucial bioenergy source widely applied in wastewater treatment. However, its efficiency improvement is hindered by complex microbial communities and sensitivity to feedstock properties. We, thus, propose a hybrid quantum-classical machine learning (Q-CML) regression algorithm using a quantum circuit learning (QCL) strategy. Combining a variational quantum circuit with a classical optimiser, this approach predicts biogas production from operational data of 18 full-scale mesophilic AD sites in the UK. Tailored for noisy intermediate-scale quantum (NISQ) devices, the low-depth QCL model outperforms conventional regression methods ( $R^2$ : 0.53) and matches the performance of a classical multi-layer perceptron (MLP) regressor ( $R^2$ : 0.959) with significantly fewer parameters and better scalability. Comparative analysis highlights the advantages of quantum superposition and entanglement in capturing intricate correlations in AD data. This study positions Q-CML as a cutting-edge solution for optimising AD processes, boosting energy recovery, and driving the circular economy.

Anaerobic digestion (AD) is a process that converts organic waste into methane, which constitutes an essential part of circular agriculture and bioeconomy<sup>1</sup>. AD process includes hydrolysis, acidogenesis, acetogenesis, and methanogenesis, reliant on microbial interactions and reactor parameters such as temperature, pH, and organic loading rate in responding to substrate and feedstock properties<sup>2,3</sup>. AD is crucial for wastewater treatment, biogas energy recovery, and digestate nutrient recycling<sup>4–6</sup>. Scalability of AD is hindered by multiple challenges, including microbial complexity<sup>7</sup>, feedstock variability<sup>2,3</sup>, low-quality biogas<sup>3,7</sup>, and digestate oversupply<sup>8,9</sup>, which lead to various advances such as co-digestion<sup>7</sup>, pretreatments<sup>8,9</sup>, and nutrient recovery methods<sup>3,9</sup> that are improving process stability and sustainability. Given the emerging applications of AD and its prominence in achieving energy output maximisation and carbon neutrality<sup>10,11</sup>, modelling of the AD process has been an active research topic in enhancing AD operators' analytical capabilities, decision-making, and planning, allowing optimisation to be made. Conventional mechanistic models, such as Anaerobic Digestion Model No. 1 (ADM1), require intricate calibration and are insufficient to capture all the physicochemical processes with each site and feedstock involved<sup>12</sup> due to the incomplete understanding of complex interactions between microbial communities that are extremely susceptible to environmental change<sup>13</sup>. Machine learning (ML) is a data-driven method that is independent of a complete knowledge to complex interactions, which

has been employed as a potential generalisable tool in improving control, operational safety, and performance forecasting in AD<sup>14</sup> due to its capability in managing multivariate data, predict nonlinear relationships, and handling incomplete datasets<sup>12,13</sup>. Traditional ML techniques including the most predominant artificial neural networks, support vector machines, and tree models have been utilised to predict biogas<sup>15</sup>, methane, hydrogen sulphide, and volatile fatty acids<sup>14</sup> productions, which are all important indicators of process stability and biogas production efficiency. Other advanced methods including deep learning and reinforcement learning have also been implemented, which are reported to be particularly beneficial for optimising the combined efficiency of feeding, heating, and mixing processes in AD<sup>14</sup>.

In this study, we propose a novel approach to modelling AD processes by applying ML techniques integrated with quantum computing (QC). QC leverages principles of quantum mechanics, such as entanglement and superposition, to process information. Unlike classical bits that can only represent 0 or 1, quantum bits (qubits) can exist in a superposition of both states simultaneously, enabling QC to handle highly complex calculations with greater efficiency. This makes quantum computing particularly promising for solving challenging problems involving high-dimensional datasets (e.g. molecules simulation and cryptography) and providing exponential speedup in some optimisation problems (e.g. micro-kinetics)<sup>16</sup>. For instance, one of the quantum algorithms, Harrow–Hassidim–Lloyd

<sup>1</sup>Department of Civil and Environmental Engineering, Imperial College London, London, UK. <sup>2</sup>Department of Electrical and Electronic Engineering, Imperial College London, London, UK. <sup>3</sup>Quantumly Ltd, Greater London, UK. <sup>4</sup>Victoria Shanghai Academy, Hong Kong SAR, China. <sup>5</sup>These authors contributed equally: Yomna Mohamed, Ahmed Elghadban. ✉e-mail: [po-heng.lee@imperial.ac.uk](mailto:po-heng.lee@imperial.ac.uk)

(HHL) algorithm<sup>16</sup> could estimate micro-kinetics in advanced oxidation process optimisation for water pollution control applications. However, most quantum algorithms, including HHL, require high-depth quantum circuits, which are difficult to implement on near-term quantum hardware systems—known as noisy intermediate-scale quantum (NISQ) devices—due to the short coherence time and inherent instability of qubits<sup>17–20</sup>. To overcome these limitations, hybrid quantum-classical ML (Q-CML) algorithms have been developed. Q-CML algorithms use classical optimisation processes to iteratively refine the parameters of low-depth quantum circuits. This approach leverages the well-established techniques of classical computational frameworks, optimisation strategies, and error correction mechanisms<sup>21–23</sup>.

This study focuses on the low-depth quantum circuit learning (QCL) algorithm within the Q-CML framework tailored for NISQ devices. QCL involves encoding classical data into qubit states using a variational quantum circuit whose parameters are optimised through classical algorithms based on the output's fidelity to expected results<sup>22</sup>. While QCL has primarily been applied to classification problems<sup>21</sup>, our research extends its utility to regression analysis, which operates similarly to a basic neural network within the quantum realm<sup>22,23</sup>, in a particular use case to model total biogas production in AD digesters as a very first step to explore the utilisation of QCL in modelling complex AD systems. To adapt low-depth QCL for regression tasks, we implement a mean squared error (MSE) loss function within the classical optimisation phase, crucial for minimising the prediction error associated with continuous outcomes. Subsequently, we apply the stochastic gradient descent (SGD) method, an optimisation algorithm renowned in classical ML for its efficiency in parameter updates. Specifically, this approach iteratively adjusts parameters in small, incremental steps based on the MSE, offering a dynamic route to loss minimisation. This method processes data in batches, leading to a globally optimised solution more efficiently than processing the entire dataset in a single iteration. The measured values from the variational quantum circuit are passed to a post-processing layer that maps these values to the range of real-world continuous outputs for NISQ devices.

In this study, we developed a QCL model from Farhi and Neven<sup>23</sup> and Schuld et al.<sup>21</sup> and performed validation using a dataset from a prior study encompassing 18 full-scale mesophilic AD sites in the UK, previously analysed through multi-level regression modelling<sup>3</sup>. This dataset allows us to compare the outcomes from the developed low-depth QCL, a classical multi-layer perceptron (MLP) approach, and previous conventional multi-level regression modelling<sup>3</sup>. This approach leverages the computational efficiency and capability of addressing regression problems offered by parameterised quantum circuits. The proposed hybrid Q-CML pipeline holds potential for application in predicting other process parameters in AD plants of a larger scale, enabling optimisation of sewage sludge AD operations.

## Results and discussion

### Feature ablation

Feature ablation is a crucial process in ML used to assess the impact of each input feature on a model's predictions. In the context of QCL, which is analogous to classical neural networks, directly extracting regression variable coefficients is not feasible. To mitigate this issue, we perform feature ablation on the model to identify significant variables and discern any that may be redundant or negatively impact the model. The QCL configuration is kept constant in this feature ablation analysis, with the specific set of hyperparameters defined in Table 1, which falls into the High Accuracy category with  $R^2 > 0.85$  achieved while utilising the full feature set. This process allows us to refine the QCL model and uncover nuanced relationships within the data. Each of the model evolution is illustrated by 3 stages: (1) including all features, (2) excluding temperature only, (3) excluding both temperature and seasonal data.

Figure 1a depicts the observed and predicted gas volume outputs of our model across three stages, encompassing all operational parameters. At each stage, the predictions generated by the QCL model align well with the

observed data, although some outliers remain unaccounted for. At Stage 1, the QCL model yields an  $R^2$  score of 0.904, whereas the  $R^2$  score increases to 0.909 when temperature data are ablated at Stage 2, suggesting that temperature data may not be as integral to the model's accuracy, since AD operating temperature is kept stable due to the need to maintain mesophilic condition for the microorganisms. Furthermore, with temperature data removed, further analysis reveals fluctuations in the  $R^2$  values post-ablation. These fluctuations indicate potential interactions between temperature and other features, leading to the hypothesis that additional feature removals could provide a clearer understanding of the model behaviour. At Stage 3, the exclusion of both seasonal and temperature data results in a more stable  $R^2$  score of 0.931. Omitting both temperature and seasonal data reduces noise and enhances the model's focus on more impactful variables for predicting biogas production. Notably, even before optimising the QCL model configuration (which is discussed in the next section), its accuracy compares favourably with the highest  $R^2$  of 0.530 reported in a previous study which utilises the superset of our chosen dataset and a conventional multi-level regression method<sup>15</sup>.

Figure 1b shows the observed and predicted gas production when the model is trained without the HRT data. The QCL model still demonstrates a good fit in the absence of hydraulic retention time (HRT) data ( $R^2$  score of 0.927), although more outlier points are observed compared with those in Fig. 4a. This suggests that the model retains a high level of predictive power without HRT due to its relatively constant values, which is typically within the stable range of 14 to 40 days for mesophilic AD<sup>24</sup>, and possibly due to the presence of other correlated operational features.

Conversely, when the model is trained without the DS feed data, as shown in Fig. 1c, the accuracy of biogas production prediction deteriorates. Excluding DS feed data from the model leads to a decreased  $R^2$  score of 0.917, a more noticeable drop compared with that in the case of the exclusion of HRT data, indicating that DS feed is a more significant predictor than HRT in the model. This reduction in accuracy corroborates the major role of substrate properties and organic loading rate, which are directly related to the DS feed, in affecting the biogas yield and process stability, aligning with our presumptions stated in the introduction.

The parameter with the most drastic effect is the site-specific simulation data. Figure 1d shows the predictions of the QCL model when trained without site data. These results are notably poor, with the  $R^2$  score dramatically decreasing to 0.020 in Stage 3. This indicates that site data, which capture site-specific conditions and operational nuances, are substantial contributors to the variability in biogas production and are critical for the model's accuracy.

Collectively, the analysis of Stage 3 results shows that the model is least impacted by the exclusion of HRT data, more affected by the exclusion of DS feed data, and cannot function effectively without site data. The substantial impact of excluding site data highlights the importance of location-specific factors in the AD process, suggesting that a one-size-fits-all model may not be adequate without considering the site-specific variables.

### Comparative analysis of QCL and MLP models

The feature importance analysis (Fig. 2a), generated using a Random Forest model, shows that learning rate accounts for more than 70% of the total importance when utilising the QCL model. The number of layers comes

**Table 1 | Hyperparameters used in comparison to previous study and feature ablation analysis**

| Hyperparameter                                   | Range of values |
|--------------------------------------------------|-----------------|
| Number of layers                                 | 4               |
| Batch size                                       | 80              |
| Learning rate                                    | 0.02            |
| Total iterations                                 | 400             |
| Train-test split ratio (Training set proportion) | 0.75            |

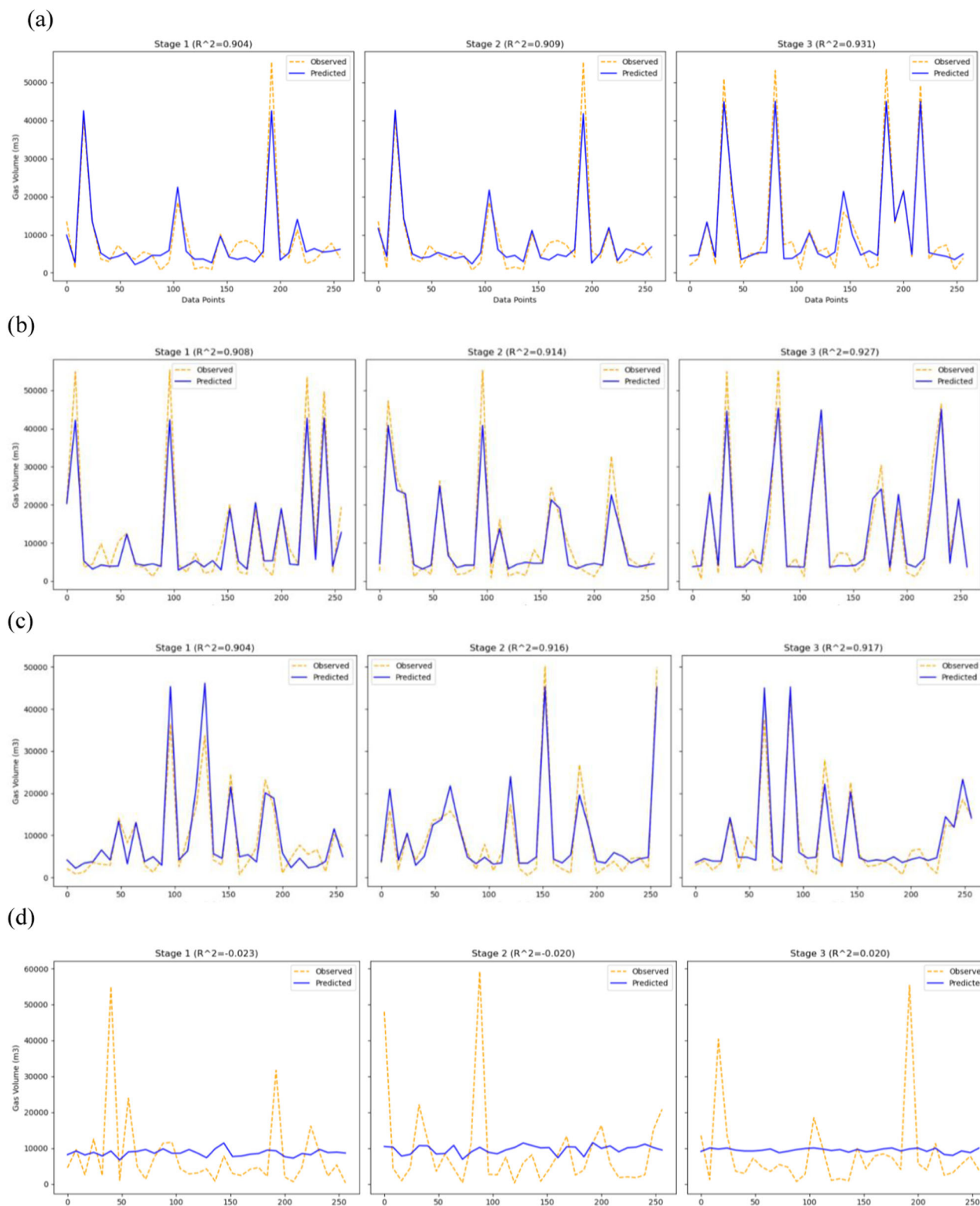


**Fig. 1 | Model performance across different stages.** Observed versus predicted gas volume outputs of the QCL model across three stages: (a) including all operational parameters (HRT, DS feed, and site specification); (b) excluding HRT; (c) excluding DS feed; and (d) excluding site specification.

second, albeit with significantly lower importance compared to the learning rate. The remaining parameters have minor influences individually. For the MLP model result shown in Fig. 2b, while the learning rate is the largest contributor (~35%), the number of layers also significantly contributes around 30% of the total importance. The global correlation heatmaps (Fig. 3a, b) between hyperparameters and model accuracy depict that the learning

rate has a correlation of 0.51 and 0.13 with the QCL and MLP model accuracies respectively, which indicates that while both models are influenced by the learning rate, QCL's accuracy is more sensitive to changes in learning rate.

To facilitate the understanding of the influence of hyperparameters on model performance, all hyperparameter configurations (experiments) are

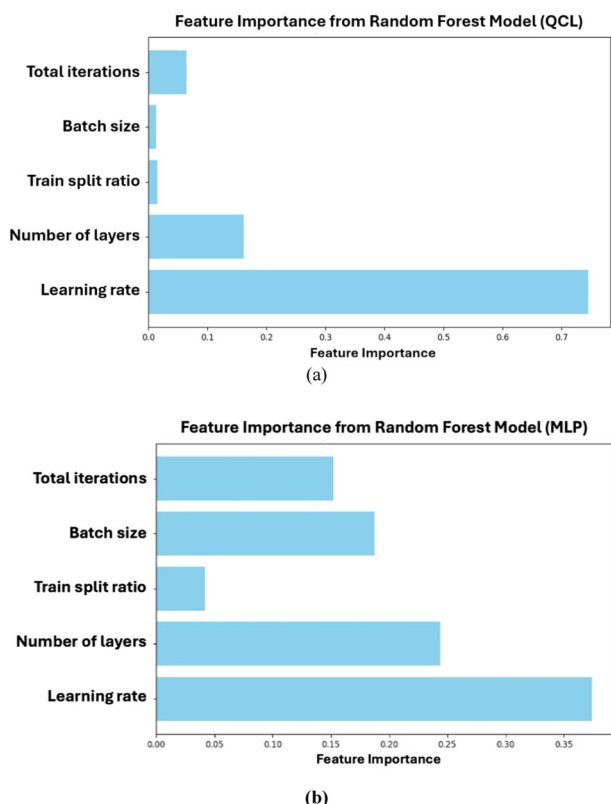


**Fig. 2 | Feature importance analysis.** Feature importance plots for (a) the QCL model and (b) the MLP model.

categorised into 3 accuracy levels based on the  $R^2$  metric. For the QCL model, 129 experiment results fall into the High Accuracy category (with  $R^2 > 0.85$ ), while 13 and 52 experiment results fall into the Medium Accuracy (with  $0.0 < R^2 \leq 0.85$ ) and Poor Fit ( $R^2 < 0$ ) categories. Statistical analysis, including ANOVA tests, confirms that these categories are significantly different from each other (ANOVA F-statistic: 189.10,  $p$ -value:  $5.13 \times 10^{-46}$ ), supporting the robustness of our categorization approach. Boxplot analyses, SHAP summary plots, and 3D interaction plots are utilised to depict the impacts of each of the 5 hyperparameters across 3 accuracy categories for both QCL and MLP models. Using the dataset (Supplementary Note 1) with QCL and MLP (Supplementary Note 2, 3, respectively), the key findings regarding learning rate and the number of layers are summarised below and the complete analysis of all 5 hyperparameters can be found in Supplementary Note 4.

For the QCL model, the distribution of learning rate values across accuracy categories is relatively concentrated, with optimal performance between 0.05 and 0.2. QCL configurations with learning rates values below 0.001 consistently fall into Poor Fit or Medium Accuracy categories. Coupled with the dominant feature importance that the learning rate holds for the QCL model, it is suggested that the learning rate value on its own directly affects the model performance, which is further supported by the result from SHAP summary plots. For the MLP model, relatively large variance in the distribution of learning rate is observed in the Medium and High Accuracy categories, indicating the learning rate affects the MLP model performance with the complement of other hyperparameters, which aligns with the relatively even distribution of feature importance of MLP model and the SHAP summary plot. The number of layers has a similar median value of around 4 layers across all accuracy categories for the QCL model, which suggests that the number of layers alone may not be a strong differentiator between accuracy categories. For the MLP model, increasing the number of layers improves model performance with observed peaks at 3 to 4 layers, beyond which the configurations mostly fall into the Medium Accuracy

category, suggesting that the effect plateaus or worsens with more than 4 layers.

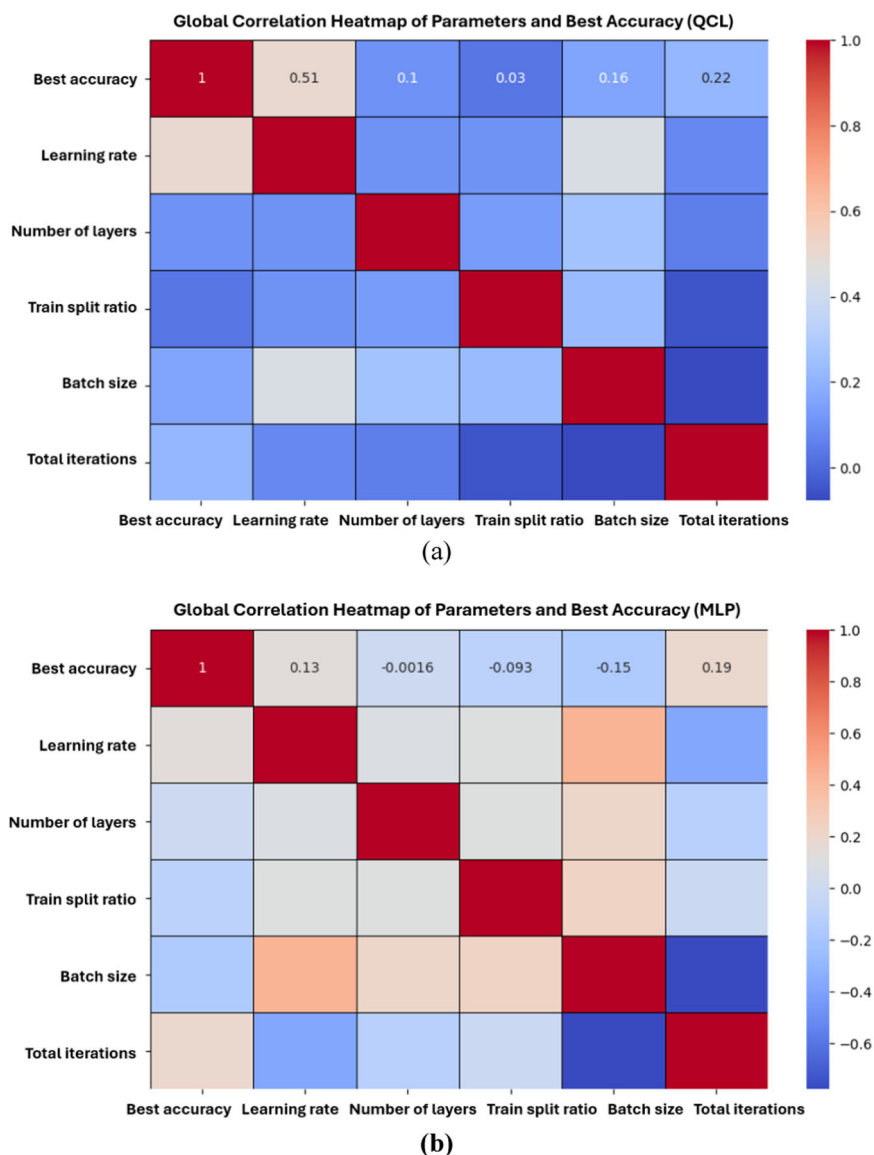
The insights from the interpretation of boxplot analyses are further investigated utilising the partial dependence plots (PDPs) between the learning rate, the number of layers, and the best model accuracy. For QCL model (Fig. 4a), the PDP for learning rate shows a steep increase in accuracy up to around 0.05, beyond which the effect plateaus. The PDP for the number of layers also shows a positive effect on QCL model accuracy up to around 6 layers, after which the effect begins to stabilise. On the contrary, the PDPs result shown in Fig. 4b from the MLP model depicts decline in accuracy when the learning rate exceeds around 0.1 and the number of layers exceeds 4. This suggests that QCL is more robust to hyperparameter changes and demonstrates a better capacity to accommodate deeper architecture, which translates to better generalisation and scalability with layers. The worsening marginal effect of the number of layers on MLP model performance is further investigated by exploring the impact of hidden layer configurations on MLP model accuracy, with the complete analysis depicted in Supplementary Note 4. While Big Nodes configurations consistently outperform Small Nodes and Mixed Nodes configurations and achieve high accuracy across 1 to 3 layers (with the highest of 0.959 with a 3-layer configuration) with low standard deviation, the variability of results associated with Big Nodes increases significantly after the number of layers exceeds 4, suggesting the possibility of overfitting and diminishing returns in deeper configurations.

Table 2 summaries a comparison of best accuracies against the corresponding total weights of the configurations provides insight into their parameter efficiency. While the MLP model attains a slightly higher best accuracy, it requires a substantially large number of weights due to its reliance on the Big Nodes configuration. The QCL model achieves comparably high accuracy with significantly fewer parameters thanks to the exploitation of superposition features in qubits.

The scatter plot (Fig. 5) for total weights versus best accuracy indicates that QCL can achieve comparable levels of accuracy with fewer total weights compared to MLP. Specifically, MLP often requires an order of magnitude more total weights to achieve similar levels of accuracy, as evidenced by the dense clustering of QCL points with significantly fewer weights. For models achieving accuracy that is higher than 90%, the MLP model (Fig. 6a) has a mean first epoch of 2.49 epochs, while the QCL model (Fig. 6b) has a mean of 2.13 epochs (translated from the number of iterations). The first best epoch is defined as the epoch at which the model achieves its highest testing accuracy, and for MLP it is 15.33 epochs on average, whereas for QCL it is 7.50 epochs. This indicates that QCL generally converges faster to its best performance compared to MLP, and it also requires fewer epochs to reach 90% accuracy initially. The convergence trends visualised in Fig. 6a show that QCL tends to have a more consistent convergence pattern compared to MLP, which exhibits higher variability. A comparison based on statistical summaries in Table 3, SHAP analysis, and PDPs reveals the differences in runtime and memory usage between the QCL and MLP models. The QCL model's runtime per epoch and memory usage during training are significantly longer than those of the MLP model. This difference is primarily due to the computational complexity involved in simulating quantum circuits on classical hardware and the need to maintaining quantum circuit state information during training. The large standard deviations for QCL also suggests variability across different hyperparameter combinations, reflecting the additional computational demands of quantum simulations.

The complete SHAP analysis and discussion regarding PDPs are shown in Supplementary Note 5, with the key findings that the number of layers and batch size are significant factors on QCL model runtime and memory usage. Whereas for the MLP model, increasing batch size reduces runtime per epoch, which is a stark contrast to QCL model. MLP model's memory usage is also less sensitive to hyperparameter changes than QCL. The QCL model's higher runtime and memory requirements indicate that, while it offers potential quantum advantages, the current computational cost is prohibitive for large-scale applications using classical simulations.

**Fig. 3 | Hyperparameter correlation with model accuracy.** Correlation heatmap showing the relationship between hyperparameters and best accuracy for (a) QCL and (b) MLP models.



However, these models hold promise for quantum hardware, where such costs may be mitigated along with future development of quantum computers.

### Statistical significance and robustness assessment

Figure 7 demonstrates High Accuracy configurations (Experiments 01 to 05, 14 to 19) using the model, exhibiting remarkable consistency across runs. The coefficient of variation (CV) for  $R^2$  is below 1%, and the 95% confidence intervals are narrow with averaged  $R^2$  values of 0.913 to 0.924 across experiments with High Accuracy configurations. Levene's test confirms homogeneity of variances, indicating that the residuals' variances are consistent across runs. Conversely, experiments using Poor Fit configurations (Experiments 06 to 13) show high variability in performance metrics, with CVs exceeding 100% in some extreme cases. The wider confidence intervals and significant results from Levene's test suggest that the model's performance was inconsistent under these configurations.

### Challenges and opportunities of Q-CML in AD applications

A potential area for enhancement and exploration in future work revolves around the process of performing measurements and obtaining expectations. The current approach runs the circuit 1000 times by default during measurements and expectation calculations. However, increasing this

number alongside upgrades in hardware capacity can provide more accurate estimates of expectation values.

In this study, the  $y$ -values are normalised to a range of  $-1$  to  $1$ . While this approach standardizes the data for computation, it introduces a granularity or "pixelation" effect, potentially distorting the finer details of the data and resulting in substantial value jumps when switching between expectations. This issue could be resolved in future work by increasing the number of runs used for calculating expectation values, as mentioned above. By aligning the number of runs more closely with the range of  $y$ -values, the pixelation effect could be reduced, thereby enhancing the precision of the expectation value calculations in our quantum computations.

Improved reproducibility and transparency are seen in water and environmental research domains due to the encouragement of open science practice<sup>25</sup>, in which researchers share data with the notion that collaborative efforts can accelerate the process of developing new solutions to enhance public utilities<sup>26</sup>. However, sensitive data from waste treatment collected at high spatial resolution are at risk of causing privacy concerns due to its ability to inform on biomarkers, potentially causing the stigmatisation of subpopulations<sup>27</sup>. The increasing importance of collecting and utilising data is inevitable, given the increasing popularity of applying machine learning and other data-driven methods in predictive modelling of public utilities. In this study, we explored the potential of exploiting the computational power

**Fig. 4 | Partial dependence of learning rate and layers.** Partial dependence plots illustrating the effects of the learning rate and number of layers on (a) QCL and (b) MLP model performance.

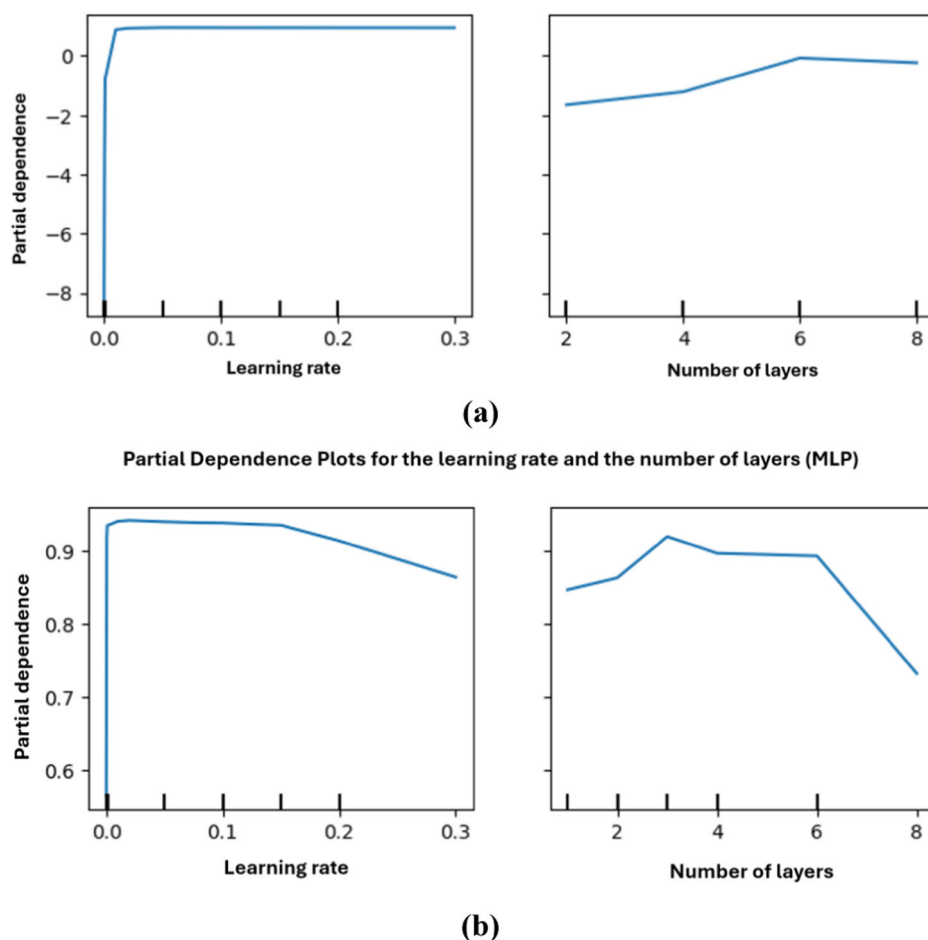


**Table 2 | Summary of key accuracies and the corresponding Weights**

| Model     | Range of weights | Best accuracy                      | Highest accuracy achieved by smallest weights |
|-----------|------------------|------------------------------------|-----------------------------------------------|
| QCL model | 28 to 100        | 0.955 achieved with 52 weights     | 0.935 achieved with 28 weights                |
| MLP model | 209 to 284,001   | 0.959 achieved with 83,001 weights | 0.949 achieved with 209 weights               |

of QC in AD to accommodate larger-scale AD operations with data collected at higher frequency and at higher geospatial resolution, as the first step to achieve real-time monitoring and optimisation of AD plants. In future development, we as researchers will utilise privacy-enhancing technologies (e.g. beacons, access control, differential privacy, and cryptography) and perform risk assessments to comply with privacy policies and ethical considerations<sup>28</sup>, while carefully balancing the usability-privacy trade-off to maximise the benefits of technological advancements to the public and the individual customers<sup>26</sup>.

#### Current achievements and future perspectives

Overall, the QCL model proves to be highly effective, outperforming conventional regression methods and exhibiting a performance comparable to that of the MLP model. Most significantly, QCL model achieves the best accuracy 0.955 with 28 weights, while MLP model achieves a slightly higher accuracy 0.959 with 83,001 weights, which demonstrates QCL model's high parameter efficiency. From the convergence analysis, QCL models demonstrate to converge faster than MLP model on average since it requires fewer epochs to reach its best performance and 90% accuracy initially. Notably, QCL also depicts better scalability with layer depth, as indicated by its stable performance with up to 8 layers, whereas MLP model performance degrades after using around 4 layers and may be more prone to overfitting.

Additionally, the conducted parameter sensitivity analysis shows that the QCL model is solely dependent on the tuning of learning rate, while MLP requires more careful tuning across multiple hyperparameters. However, it is important to acknowledge that the current computational cost and run-time required for QCL models on classical computers are extremely more expensive than utilising MLP models, owing to the complexity and memory involved in simulating quantum circuits during training. While QCL models demonstrates higher parameter and training efficiency and better scalability than MLP models in this analysis, its real-world application holds promise and highly depends on the future development of quantum hardware.

The results verify and align with a series of presumptions regarding both AD processes and the applied QCL model. They corroborate the primary role of substrate properties (represented by TDS, DS feed, and total digester volume) in affecting the overall AD performance while being an indispensable feature in the modelling process. The strong entanglement feature created by low-depth quantum circuit architecture is verified to effectively facilitate the performance of regression tasks, which suggests that QCL models may be beneficial in capturing intricate correlations between input features when compared to classical ML models. In addition, the results provide insights into the necessity to incorporate location-specific or site-specific information to ensure the performance of the

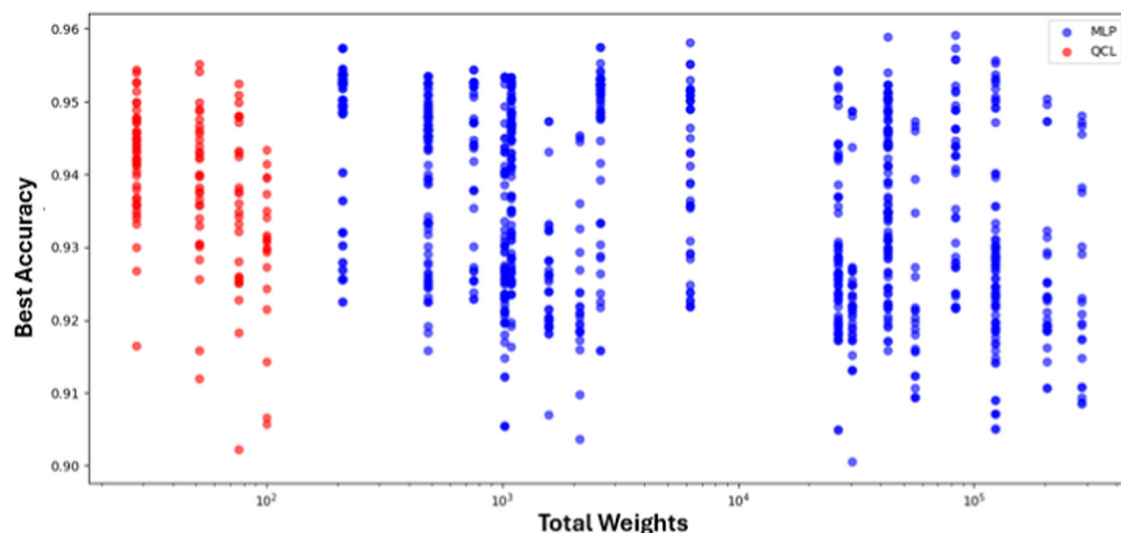


**Fig. 5 | Model weights vs. accuracy.** Total weights plotted against the best accuracy for QCL and MLP models, showing only accuracies higher than 95%.

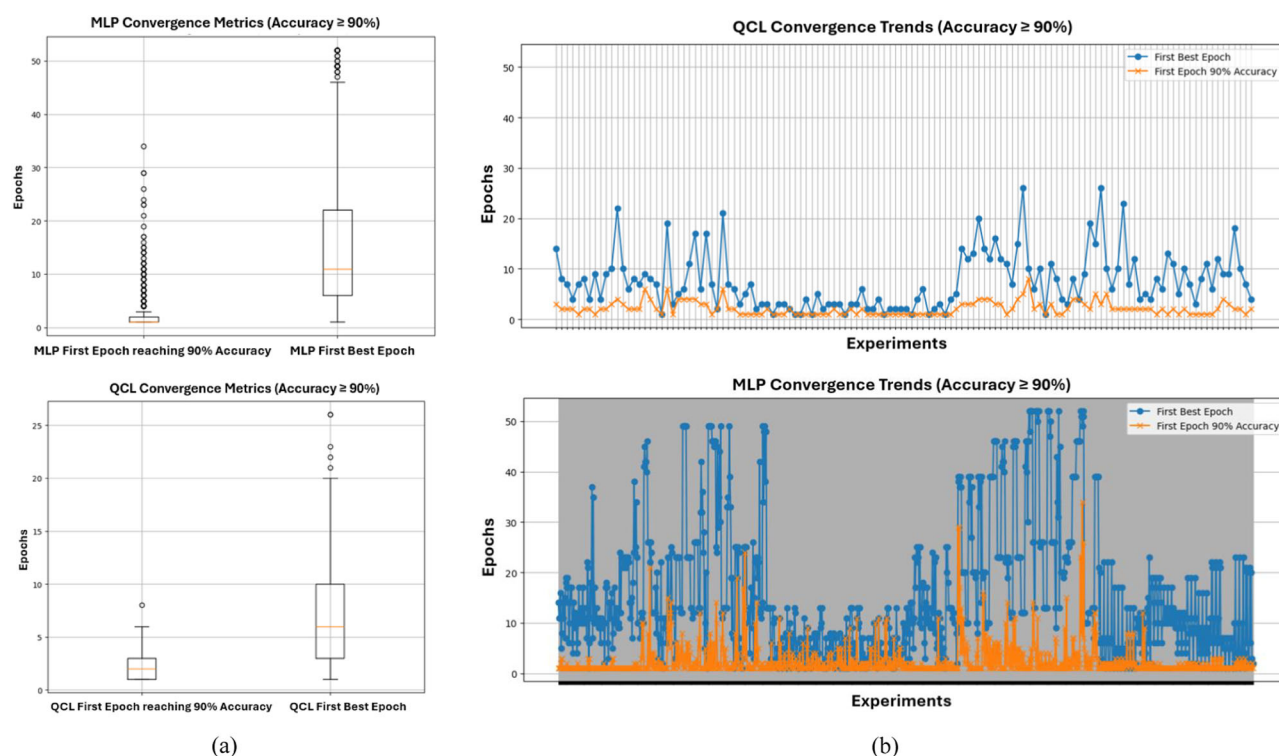


**Fig. 6 | Convergence analysis of model configurations.** Convergence metrics (a) and convergence trends (b) for MLP and QCL configurations achieving at least 90% accuracy.

**Table 3 | Summary of runtime and memory usage for QCL and MLP models**

| Analysis                                     | Model | Mean runtime (s)       | Median runtime (s)       | Standard deviation (s)  |
|----------------------------------------------|-------|------------------------|--------------------------|-------------------------|
| Runtime analysis (epoch-to-epoch comparison) | QCL   | 73.69                  | 63.32                    | 54.70                   |
|                                              | MLP   | 0.07                   | 0.04                     | 0.11                    |
| Analysis                                     | Model | Mean memory usage (MB) | Median memory usage (MB) | Standard deviation (MB) |
| Memory usage analysis                        | QCL   | 2875.04                | 2571.05                  | 950.58                  |
|                                              | MLP   | 234.12                 | 241.38                   | 31.27                   |

proposed Q-CML framework in modelling process parameters and outputs in AD plants.

This study can be regarded as a very first step to exploit QCL's power in the predictive modelling of AD process parameters, which provides an essential foundation for large-scale optimisation of industrial AD plants and potentially facilitates real-time monitoring and decision-making of AD operators. Given AD's centrality in waste management and renewable energy, the effectiveness and efficiency in modelling this highly variable, sensitive, and complex biological treatment process need to be improved via novel and powerful computational tools. QCL has the potential to fill this gap, which has remarkable implications in both environmental and energy sectors and reinforces AD's role in the circular bioeconomy.

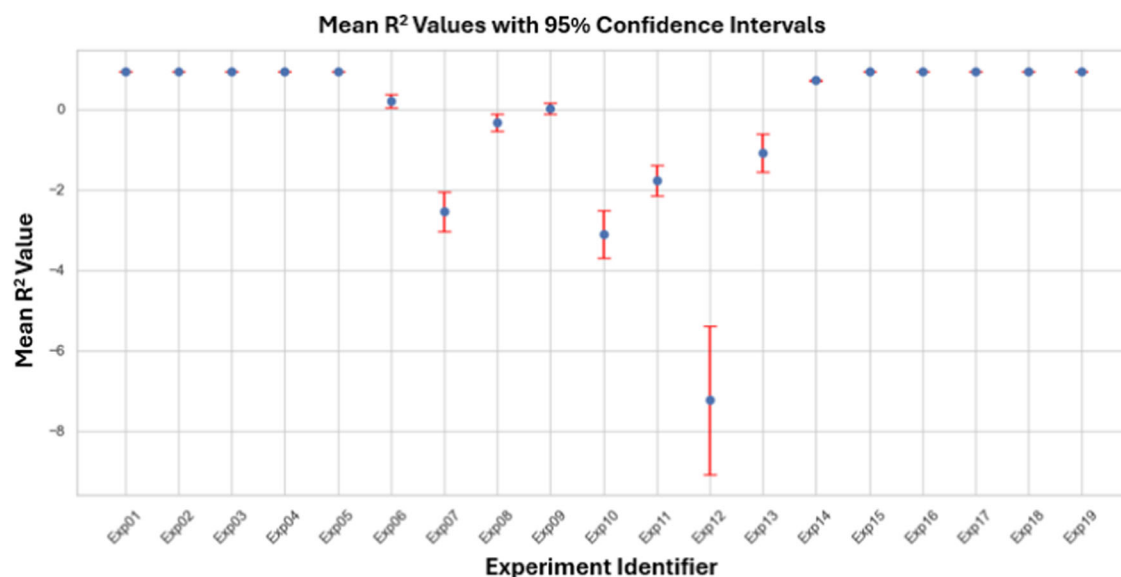


Fig. 7 | Model performance with confidence intervals. Mean  $R^2$  values for the models with 95% confidence intervals.

## Methods

### Dataset characterisation: site and seasonal analysis

The dataset used is a subset of the comprehensive operational data analysed by Liu et al.<sup>15</sup>, covering records from 66 conventional and 6 thermal hydrolysis process (THP) mechanical autothermal digestion (MAD) sludge treatment facilities in the UK, spanning the period from 2009 to 2017. This extensive dataset includes critical operational parameters, such as digestion temperature ( $^{\circ}\text{C}$ ), HRT (days), and percentage of dry solids (DS) in the sludge feed (%), with outputs quantified in terms of biogas volume ( $\text{Nm}^3$ ) and biogas yield ( $\text{m}^3\text{t}^{-1}\text{DS}$ ). For our focused analysis, we selected data from 18 facilities, spanning from 2011 to 2017, as these sub-datasets provide the most continuous and complete data with a high degree of data integrity and consistency.

In this study, we focus on examining the total volume of biogas produced. To further enhance the robustness of our model, we incorporate two supplementary features: the total volume of digester feed and the total volume of dissolved solids (TDS), which form an 11-feature dataset together with site-specific data. The influence of these features, in particular DS Feed and HRT, on the biogas production process is examined, given their significant impact on the microbiological activity within the digesters, and their utility in improving the predictive accuracy of the proposed QCL regression model is explored. The dataset characterisation in terms of seasonal and site-specific variability is presented in Supplementary Note 1, 2.

### Model development and implementation

Figure 8a illustrates the flow of the hybrid Q-CML model, which consists of five elements: data initialisation, quantum circuit, measurement of observables, cost function minimisation, and performance evaluation. Figure 8b shows the detailed simulation flow of the proposed QCL framework. We use parameterised quantum circuits, of which the parameters are optimised classically during the learning process. The quantum circuit functions as a universal function approximator, mapping the encoded input to the desired output. The observables are measured at the end of the quantum circuit and the expectation values are passed onto the classical postprocessing layer and subsequently the optimisation layer. The classical optimisation element (SGD) updates the circuit parameters, leveraging the capacity of variational circuits to efficiently compute gradients.

The initialisation step transforms classical information into the quantum domain by encoding the feature vector, or the input data point, into a quantum state of  $n$  initialised qubits. For  $n$  qubits, amplitude encoding can represent  $2^n$  classical features. This efficiency is achieved by exploiting

the exponential state space and superposition property of a binary quantum system, which means that a qubit can exist in multiple states simultaneously<sup>22</sup>. By reducing the number of required qubits, amplitude encoding also aligns well with the constraints of NISQ devices, making it feasible to process larger datasets on current quantum hardware. This efficiency has been demonstrated in prior studies as a practical advantage, especially in resource-limited quantum systems<sup>22,29</sup>. Note that this procedure assumes that the input data can be represented as a probability distribution because the quantum states are represented as normalised vectors in a Hilbert space. More generally, when the number of features is less than  $N$ , the feature vector is padded with additional features. More details on amplitude embedding can be found in existing work<sup>30,31</sup>.

One critical requirement of amplitude encoding is the horizontal normalisation of each input datapoint by row. This ensures that the total probability associated with the encoded quantum state sums to one, which is necessary for quantum state preparation. Mathematically, this normalisation is achieved by transforming each input feature vector  $x$  such that its normalised components satisfy  $\sum_{i=1}^N x_{i-\text{normalised}}^2 = 1$ . This process ensures that the prepared quantum state adheres to the unitary requirements of quantum computing and accurately reflect the data point's structure<sup>32</sup>. Notably, this normalisation procedure assumes that all input data points are non-negative and real-valued. More generally, if the original data do not satisfy these conditions, suitable pre-processing or transformation methods should be applied before amplitude encoding. Subsequently, the elements in the  $N$ -feature vector  $x$  are normalised as follows (Eq. 1):

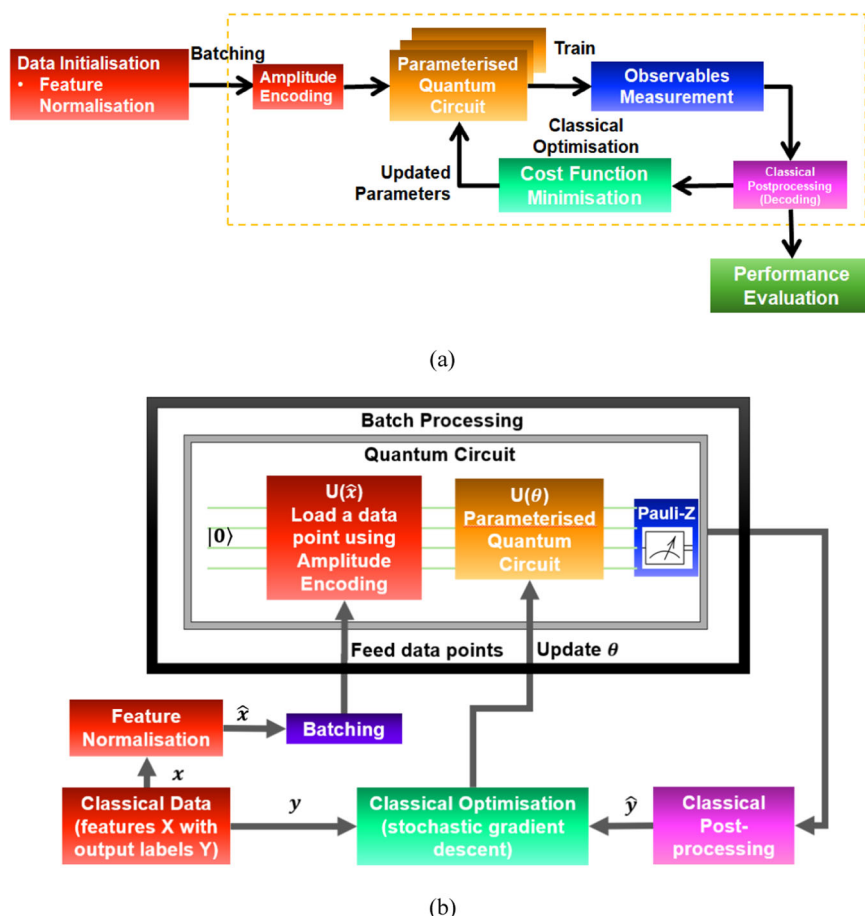
$$x_{i-\text{normalised}} = \frac{x_i}{\sqrt{\sum_{i=1}^N (x_i)^2}} \quad (1)$$

where  $x_{i-\text{normalised}}$  represents the normalised value of each input feature (e.g. HRT, digestion temperature, DS) in one feature vector, and  $x_i$  represents the input feature value before normalisation.

Amplitude encoding is resource-efficient but requires precise state preparation, making it sensitive to noise and prone to errors that can degrade algorithm performance<sup>29</sup>. Additionally, its emphasis on feature magnitudes may amplify biases, affecting the model's generalisation<sup>33</sup>. To address these issues, we incorporated feature (vertical) normalisation as a preprocessing step before dataset splitting. This step ensures that each feature is scaled independently across all datapoints, preventing dominant features from disproportionately influencing the encoding. Vertical normalisation effectively complements the horizontal normalisation intrinsic to

**Fig. 8 | Hybrid Q-CML model framework.**

**a** Process flow diagram of the hybrid Q-CML model, highlighting five key elements: data initialisation, quantum circuit, observable measurement, cost function minimisation, and performance evaluation. **b** Detailed simulation flow for the proposed QCL implementation.



amplitude encoding by providing a more balanced dataset representation. This preprocessing ensures that the encoded quantum state reflects the dataset's overall structure, reducing the risk of bias and improving the model's ability to generalise<sup>34</sup>. Detailed information of vertical normalisation can be found in Supplementary Note 1.

The following simplified example (Eqs. 2–4) of a 4-feature vector  $x = [x_1, x_2, x_3, x_4] \forall x_i \in \mathbb{R}_+$  further illustrates the implementation of amplitude encoding, which is also the preparation of qubit states before being inputted into the parameterised quantum circuit:

Step 1: Initialise quantum state  $|\psi\rangle$

Since there are  $N = 4$  features, only 2 qubits are required in compiling with  $N \leq 2^n$ . In other words, there are 4 possible states for the overall quantum system since each qubit can be in one of the 2 computational basis states, namely state 0 or state 1, denoted by  $|0\rangle$  and  $|1\rangle$  respectively.

The general state of a 2-qubit system can be represented by the following wave function, which is a superposition of 4 possible states:

$$|\psi\rangle = a_1|00\rangle + a_2|01\rangle + a_3|10\rangle + a_4|11\rangle \quad (2)$$

where we aim to encode the feature vector  $x$  in the probability amplitudes  $a_i \forall i \in [1, 4]$ .

Step 2: Normalise feature vector  $x$

The unitary requirement in quantum mechanics demands  $\sum_{i=1}^4 (a_i)^2 = 1$ . Hence, we normalise  $x$  to facilitate the encoding process.

$$x_{\text{normalised}} = \frac{1}{\sqrt{\sum_{i=1}^4 (x_i)^2}} * [x_1, x_2, x_3, x_4] \quad (3)$$

Step 3: Encoding feature vector  $x$  into quantum state  $|\psi\rangle$

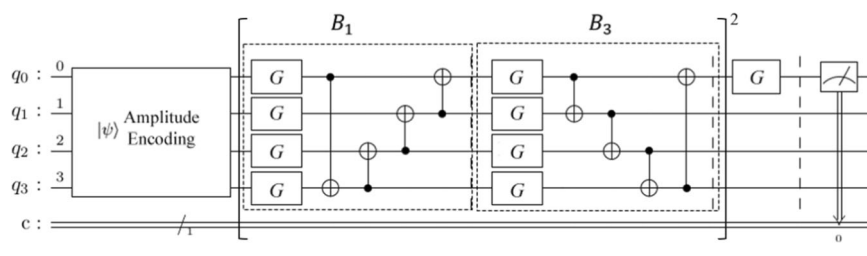
$$|\psi\rangle = \frac{1}{\sqrt{\sum_{i=1}^4 (x_i)^2}} * [x_1|00\rangle + x_2|01\rangle + x_3|10\rangle + x_4|11\rangle] \quad (4)$$

The same framework is used in the amplitude encoding of a feature set which comprise classical information including HRT, temperature, digester feed total volume, DS, TDS, seasonal data, and other features related to the characteristics of the individual AD plant. A 4-qubit system is used for the experiments and directly extends this simplified example.

The QCL circuit is developed with reference to the work of Schuld et al.<sup>21</sup>. Figure 9 demonstrates a layered circuit structure in the algorithm implementation. Each layer consists of a rotation gate (a single-qubit gate,  $G$  gate) for each qubit, with the set of rotation angles  $\theta$  being a circuit parameter which is initialised randomly using normal distribution and optimised classically, followed by controlled-NOT (CNOT) gates to entangle one qubit with another and to form a strongly entangled circuit. Entanglement is a unique property of quantum systems and creates a link between qubits, establishing connections that persist regardless of distance, where a qubit cannot be described independently of the state of the other qubits<sup>20</sup>.

The rationale for utilising the entanglement feature lies in designing low-depth circuits that align with NISQ-era hardware constraints. While low-depth circuits limit access to the full Hilbert space and reduce model flexibility<sup>35</sup>, utilising strongly entangled quantum states enhances the model's ability to capture complex correlations, both long- and short-range, in input features, thereby increasing its expressiveness<sup>36,37</sup>. QCL, akin to a quantum neural network with unitary layers<sup>22,35</sup>, requires a strongly entangled circuit architecture to capture the intricate correlations in AD processes, which are highly sensitive to environmental changes and complex

**Fig. 9 | Quantum circuit design for QCL framework.** Design of the quantum circuit as part of the QCL framework implementation.



microbial interactions. Schuld et al. demonstrated that a cyclic code block architecture not only creates strong qubit entanglement but also prevents overfitting due to unitarity and enhances noise resilience, maintaining the QCL model's generalisability<sup>21,22</sup>.

Within this generic method, we incorporated the number of quantum circuit layers as a hyperparameter, owing to the limitation of quantum circuit depth for better adaptability to the NISQ era's hardware limitations in the interest of future applications on an actual quantum hardware. In the implementation shown in Fig. 2, the circuit design has a total of 4 code blocks at depth 21 comprising 33 gates when 2 layers are employed, with a control proximity range (i.e. the number of qubits that the control crosses to reach the target)  $r \in \{1, 3\}$ . This is based on Schuld et al.'s observation<sup>35</sup> that the degree of entanglement could be significantly improved if  $r$  is relatively prime to the number of qubits  $n$ . Hence, there are two types of blocks ( $B_1$  and  $B_3$ ) in our chosen circuit design, corresponding to  $r = 1$  and  $r = 3$  respectively. Detailed equations regarding the cyclic code block structure are presented in Supplementary Note 2.

Overall, when viewing all the code blocks as a single quantum circuit, the initially encoded quantum state is subject to a  $\theta$ -parameterised unitary  $U(\theta)$ , transforming the input state to an output state, as indicated in Eq. 5.

$$|\psi_{out}(x_i, \theta)\rangle = U(\theta)|\psi_{in}(x_i)\rangle \quad (5)$$

where  $\theta$  is the circuit parameter set comprising of rotation angles,  $|\psi_{in}\rangle$  is the input state formed by the encoded input features, and  $|\psi_{out}\rangle$  is the resultant output state following the transformation of the input state by the unitary operator  $U(\theta)$ .

The expectation values of the output states  $|\psi_{out}\rangle$  are measured. Specifically, we use a subset of Pauli operators:  $B_j \subset \{I, X, Y, Z \otimes N\}$ . An output function  $F$  is introduced to define the output  $y_i = y(x_i, \theta)$  as shown in Eq. 6:

$$y(x_i, \theta) = F(\{B_j(x_i, \theta)\}) \quad (6)$$

where  $B_j$  represents quantum logic gates, e.g.  $I$  (identity gate) and  $X, Y, Z$  (Pauli  $X, Y$ , and  $Z$  gates, respectively);  $y$  is the expectation value of the output; and  $\theta$  is the parameter set updated by the classical optimiser. In our implementation, the Pauli  $Z$  gates are used for measurements since the qubits are initialised using the computational basis states, also known as orthogonal  $z$ -basis states. The utilisation of Pauli  $Z$  gates projects the transformed qubit state onto one of the computational basis states, which enables the expectation values (related to probability amplitudes) of the quantum state to be evaluated by repeated applications and measurements of the overall circuit on the simulator platform.

### Cost function minimisation

To effectively optimise iteration performance in the SGD step, we develop an equation (Eq. 7) that incorporates the de-normalisation of the range of the normalised expectation values, despite the inherent randomness. This iterative approach in SGD eventually converges, yielding an optimised set of parameters for our quantum circuits. The measured expectation value  $y_i$  is considered a normalised value of  $y'_i$ , which is then converted back to its

original range prior to normalisation:

$$y'_i = \left( (y_i + 1) \times \frac{\max(f(x_i)) - \min(f(x_i))}{2} \right) + \min(f(x_i)) \quad (7)$$

In this phase of the QCL regression process, the objective is to minimise a cost function, the MSE in this case, by iteratively fine-tuning the parameters  $\theta$  of the quantum circuit. This cost function represents the difference or error between the desired output, specified by a teacher function  $f(x_i)$  based on the range of the sample dataset, and the actual output  $y_i$  produced by the quantum circuit. SGD directs us toward the steepest increase in the cost function, and to minimise it, we update the parameters by taking a step in the opposite direction. The size of this step is determined by the learning rate hyperparameter.

### Simulation platform

We use PennyLane's quantum computer simulators over actual quantum computers due to logistical considerations and computational efficiency. Due to the limited number of actual NISQ quantum computers available for research, classical simulators effectively serve as an initial stage for developing and testing quantum algorithms before implementing them on real quantum hardware<sup>20</sup>. Additionally, the QML optimisation approach adopted in the study involves SGD, requiring numerous iterations and data batch divisions. This necessitates running thousands of circuits per iteration. The lengthy queues associated with actual quantum computers would make it unrealistic to execute a reasonable number of experiments within a practical timeframe. Therefore, simulators are used as they can handle such tasks more efficiently, thereby streamlining the workflow and reducing the total execution time. Future work could benefit from securing quota access to IBM's Quantum Computers to bypass the queue, further enhancing the fidelity of our research.

The code is built using the pennylane.ai library, selected considering its extensive benefits, including multi-platform compatibility and integration with classical ML libraries. PennyLane offers compatibility with various quantum computer cloud platforms, such as Google and IBM<sup>38</sup>, broadening potential access to quantum hardware. Given that QCL and near-term quantum computer algorithms require classical optimisation, PennyLane's strong integration with TensorFlow and PyTorch is a critical factor<sup>38</sup>, which allows for the exploitation of PyTorch's optimisation functionalities and provides an obvious advantage than its counterparts.

### Data pre-processing

To accommodate the limitations of current quantum computers and simulators regarding qubit availability, the datasets are meticulously pre-processed to convert non-numeric columns to numeric formats and minimise the feature space.

The dataset, originating from various sites engaged in gas production from sewage AD, initially underwent a phase of null-value imputation. A k-Nearest Neighbours (kNN) imputer was employed to address missing values, adhering to the principle of maintaining data integrity and minimising the introduction of bias. The choice of kNN was guided by the specific characteristics of the dataset, including its mixed data types, non-linear dependencies, and temporal structure.

**Table 4 | Statistical results of the conventional and THP MAD datasets**

| Variable                   | <i>t</i> -statistic | <i>p</i> -value | Cohen's <i>d</i> |
|----------------------------|---------------------|-----------------|------------------|
| Digester feed total volume | 0.032558808         | 0.974032468     | 0.000918032      |
| DS feed                    | 1.499299892         | 0.134096727     | 0.05622496       |
| Digester feed TDS          | 2.428179335         | 0.01534425      | 0.10301609       |
| HRT                        | -0.63680817         | 0.52438954      | -0.025023216     |
| Temperature                | 1.757180543         | 0.079175709     | 0.068820137      |

The dataset comprised both continuous features, such as the total volume of digester feed, temperature, and total biogas produced, and categorical features like season and sitename. kNN imputation is particularly effective in such cases, as it estimates missing values based on the similarity of observations across multiple features, preserving the inherent relationships in the data. Unlike simpler methods such as mean or median imputation, kNN leverages local patterns by identifying the kNN of a data point and imputing missing values as a function of their values. This ensures that imputed values align with the underlying structure of the data.

Another critical factor influencing the choice of kNN was its adaptability to nonlinear relationships. Process variables in this dataset, such as HRT and temperature, exhibit potential nonlinear dependencies that simpler imputation methods cannot account for. Recent advancements in kNN imputation have demonstrated its effectiveness in handling complex patterns in data, particularly in scenarios involving nonlinear and heterogeneous relationships<sup>39,40</sup>. By leveraging a distance metric to identify neighbours, kNN captures local variations while preserving broader data trends, ensuring that imputation does not distort the dataset's statistical properties.

The temporal and contextual structures within the dataset also played a significant role in selecting kNN. Variables such as season, HRT, and DS feed are interconnected in ways that reflect process dynamics. kNN imputation, by considering feature similarity, preserves these interdependencies and prevents the introduction of artifacts that could disrupt downstream analyses. This characteristic is particularly advantageous for datasets in industrial and process-driven domains, where relationships between variables are critical for accurate modelling and interpretation<sup>41</sup>.

Finally, the parameter *k*, representing the number of neighbours, was optimised through cross-validation. A value of *k* = 5 was chosen to strike a balance between capturing local variability and minimising noise. Larger values of *k* can oversmooth the imputed data which obscures subtle patterns, while smaller values can amplify noise. This tuning step ensured that the imputation process was both robust and reliable, maintaining the dataset's integrity for subsequent analyses.

To study the effect of the imputation process on the dataset, a paired *t*-test is conducted, and Cohen's *d*, a metric indicating the extent of the effect of a process, is calculated (Table 4). The paired *t*-test evaluates the significance of any statistical difference in the mean of a variable before and after pre-processing, while Cohen's *d* quantifies the magnitude of this difference, providing insights into the effect size of our pre-processing procedures. Table 4 summarises the statistical results of the conventional and THP MAD datasets. A *p*-value below 0.05 indicates a significant pre-processing impact. Cohen's *d* values (Table 4) are classified as large effect size ( $\geq 0.8$ ), medium effect size ( $\geq 0.5$ ), small effect size ( $< 0.5$ ), and negligible effect size ( $< 0.2$ ). Aggregating the results for each variable, the paired *t*-test outcomes reveal no significant change post-pre-processing for the digester feed total volume, dry solid (DS) feed, HRT, and temperature, with negligible to small effect sizes. However, the digester feed TDS exhibits a statistically significant change post-pre-processing, albeit with a small effect size. Overall, these findings suggest minimal impact of pre-processing on most variables.

To enable quantum algorithm simulations, categorical variables such as sites and seasons are converted into numerical values through a pivoting

**Table 5 | Hyperparameters and their range of values used in QCL sensitivity analysis**

| Hyperparameter                                   | Range of values           |
|--------------------------------------------------|---------------------------|
| Number of layers                                 | From 2 to 8 layers        |
| Batch size                                       | From 16 to 128            |
| Learning rate                                    | From 0.0001 to 0.3        |
| Total iterations                                 | From 80 to 500 iterations |
| Train-test split ratio (training set proportion) | From 0.6 to 0.8           |

process. This step helps optimise the feature space to enhance the quantum computation efficiency.

To align with the constraints of QC, a rigorous dimensionality reduction process is conducted. Non-essential features, particularly those contributing minimal information gain, are systematically removed. This step is critical in reducing the number of required qubits, making the dataset more amenable to the current capacities of quantum computers and simulators. The consolidation of various site-specific data points into a singular "Other site" metric is a notable aspect of this reduction. By summing the values from multiple lesser-impact sites, we effectively condense the information while retaining its relevance and significance for the analysis. The concluding phase involves the removal of temporal variables, specifically "Year" and "Month," which are deemed redundant for the core objectives of this study. The resulting dataset, primarily composed of numerically transformed site data, seasonality indicators, and key operational metrics is then filtered to exclude any instances where the dependent variable ("Total biogas produced") is missing.

### Parameter sensitivity analysis

Hyperparameter tuning is crucial for optimising machine learning models, particularly in complex architectures like QCL models. QCL poses unique challenges due to their intricate parameter spaces and the nascent understanding of suitable circuit architectures for specific tasks<sup>42</sup>. Our objective is to evaluate the influence of hyperparameters, including learning rate, number of layers, batch size, train-test split ratio, and total iterations, on the QCL model's accuracy. We generated 194 distinct combinations of hyperparameter settings, representing comprehensive coverage of the parameter space. The settings cover a wide range of values for each parameter, as shown in Table 5. The distribution and coverage of these hyperparameters across the parameter space are visualised in Supplementary Note 4, showing how the selected values span the different dimensions of the hyperparameter space.

There are fundamental differences between QCL and MLP models regarding training methodology and layer configuration. While QCL model utilises iterations (ranges from 80 to 500) which process a randomly selected batch from the training data at once, the MLP model utilises epochs which represents a full pass through the entire training dataset. Hence, the number of epochs used in the MLP model training ranges from ~7 to 52, corresponding to the total iterations in the QCL model for a fair comparison. Two key strategies, namely layer mapping and hidden layer configurations, are adopted for layer configuration in MLP model due to the fundamental differences in model architecture of both models. In the QCL model, a layer is composed of a rotation gate (incorporating the optimised weights) on each qubit followed by circular entanglement gates. This differs significantly from the structure of MLP layers, which is further explained in Supplementary Note 3.

For each of the two-layer mapping configurations in Table 6, there are three hidden layer configurations that are adopted respectively. This results in 6 different combinations of layer configurations which are then evaluated with each of the 194 QCL hyperparameter combinations, resulting in 1164 configurations for the MLP model in the comparative sensitivity analysis. This approach ensured that the MLP model's hyperparameter space closely mirrored that of the QCL model, allowing for a meaningful comparison between the two models.

**Table 6 | Key strategies for layer configuration of MLP model in sensitivity analysis**

| Layer mapping |                                                                  | Hidden layer configurations |                                                                                                     |
|---------------|------------------------------------------------------------------|-----------------------------|-----------------------------------------------------------------------------------------------------|
| Full layers   | The MLP model uses the same number of layers as the QCL model    | Big nodes                   | More than 200 nodes per layer, providing more expressiveness                                        |
|               |                                                                  | Small nodes                 | Less than 16 nodes per layer for a less complex representation                                      |
| Half layers   | The MLP model uses half of the number of layers as the QCL model | Mixed nodes                 | Various number of nodes in between 16 and 200 to evaluate the effect of heterogeneity across layers |

### Statistical significance and robustness assessment

To evaluate the statistical significance and robustness of the QCL model, a series of experiments using the following techniques has been conducted to capture the inherent variability introduced by different sources of randomness.

The random initialisation of circuit weights is represented by each quantum gate's rotation angles, which are initialised for every run, introducing variability in the starting point of the optimisation process. Random seeds were set differently in each run across all libraries involved, including data loading, preprocessing, and model training functions. This practice ensures that the randomness is not constrained by fixed seeds, providing a more realistic evaluation of the model's robustness (Supplementary Note 5). Additionally, 10 independent runs for each hyperparameter configuration to assess consistency, enabling the observation of the fluctuation in model performance per experiment (Supplementary Note 5).

In each run, key performance metrics including  $R^2$ , MSE and mean absolute error are measured to quantify the model's predictive accuracy. Analysing these metrics across runs and experiments enabled us to assess both the average performance and the variability associated with each hyperparameter configuration. A range of hyperparameter settings, including both High Accuracy (with  $R^2 > 0.85$ ) and Poor Fit (with  $R^2 < 0$ ) configurations, are selected to examine the model's behaviour under both optimal and poor conditions and to identify factors contributing to performance variability and instability.

### Data availability

The dataset used in this study was provided by Thames Water Utilities Limited and can be made available upon request.

### Code availability

The code can be made available upon request.

Received: 31 August 2024; Accepted: 25 January 2025;

Published online: 05 March 2025

### References

- Li, X. et al. A comprehensive review of the strategies to improve anaerobic digestion: their mechanism and digestion performance. *Methane* **3**, 227–256 (2024).
- Meegoda, J. N., Li, B., Patel, K. & Wang, L. B. A review of the processes, parameters, and optimization of anaerobic digestion. *Int. J. Environ. Res. Public Health* **15**, 2224 (2018).
- Uddin, M. M. & Wright, M. M. Anaerobic digestion fundamentals, challenges, and technological advances. *Phys. Sci. Rev.* **8**, 2819–2837 (2023).
- Khoshnevisan, B. et al. A critical review on livestock manure biorefinery technologies: sustainability, challenges, and future perspectives. *Renew. Sustain. Energy Rev.* **135**, 110033 (2021).
- Mancini, G. et al. A water-waste-energy nexus approach to bridge the sustainability gap in landfill-based waste management regions. *Renew. Sustain. Energy Rev.* **137**, 110441 (2021).
- Kabeyi, M. J. B. & Olanrewaju, O. A. Biogas production and applications in the sustainable energy transition. *J. Energy* **2022**, 8750221 (2022).
- Chen, Y., Cheng, J. J. & Creamer, K. S. Inhibition of anaerobic digestion process: a review. *Bioresour. Technol.* **99**, 4044–4064 (2008).
- Tawfik, A. et al. Methods to alleviate the inhibition of sludge anaerobic digestion by emerging contaminants: a review. *Environ. Chem. Lett.* **20**, 3811–3836 (2022).
- Akor, C. I. et al. Thermokinetic study of residual solid digestate from anaerobic digestion. *Chem. Eng. J.* **406**, 127039 (2021).
- Osman, A. I. et al. Coordination-driven innovations in low-energy catalytic processes: Advancing sustainability in chemical production. *Coord. Chem. Rev.* **514**, 215900 (2024).
- Inyang, V., Laseinde, O. T. & Kanakana, G. M. Techniques and applications of lignocellulose biomass sources as transport fuels and other bioproducts. *Int. J. Low-Carbon Technol.* **17**, 900–909 (2022).
- Andrade Cruz, I. et al. Application of machine learning in anaerobic digestion: perspectives and challenges. *Bioresour. Technol.* **345**, 126433 (2022).
- Ling, J. Y. X. et al. Machine learning methods for the modelling and optimisation of biogas production from anaerobic digestion: a review. *Environ. Sci. Pollut. Res.* **31**, 19085–19104 (2024).
- Rutland, H., You, J., Liu, H., Bull, L. & Reynolds, D. A systematic review of machine-learning solutions in anaerobic digestion. *Bioengineering* **10**, 1410 (2023).
- Liu, J. & Smith, S. R. A multi-level biogas model to optimise the energy balance of full-scale sewage sludge conventional and THP anaerobic digestion. *Renew. Energy* **159**, 756–766 (2020).
- Harrow, A. W., Hassidim, A. & Lloyd, S. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.* **103**, 150502 (2009).
- Schuld, M., Sinayskiy, I. & Petruccione, F. An introduction to quantum machine learning. *Contemp. Phys.* **56**, 172–185 (2015).
- Corcoles, A. D. et al. Challenges and opportunities of near-term quantum computing systems. *Proc. IEEE* **108**, 1338–1352 (2020).
- Mavadia, S., Frey, V., Sastrawan, J., Dona, S. & Biercuk, M. J. Prediction and real-time compensation of qubit decoherence via machine-learning. *Nat. Commun.* **8**, 14106 (2017).
- Cho, A. The biggest flipping challenge in quantum computing. *Science*. <https://doi.org/10.1126/science.abd7332> (2020).
- Schuld, M., Bocharov, A., Svore, K. M. & Wiebe, N. Circuit-centric quantum classifiers. *Phys. Rev. A* **101**, 032308 (2020).
- Mitarai, K., Negoro, M., Kitagawa, M. & Fujii, K. Quantum circuit learning. *Phys. Rev. A* **98**, 032309 (2018).
- Farhi, E. & Neven, H. Classification with quantum neural networks on near term processors. Preprint at <https://arxiv.org/abs/1802.06002> (2018).
- Srisowmeya, G., Chakravarthy, M. & Nandhini Devi, G. Critical considerations in two-stage anaerobic digestion of food waste – A review. *Renew. Sustain. Energy Rev.* **119**, 109587 (2020).
- Zipper, S. C. et al. Balancing open science and data privacy in the water sciences. *Water Resour. Res.* **55**, 5202–5211 (2019).
- Ruddell, B. L. et al. Guidance on the usability-privacy tradeoff for utility customer data aggregation. *Util. Policy* **67**, 101106 (2020).
- Driver, E. M. et al. Encrypted data-sharing for preserving privacy in wastewater-based epidemiology. *Sci. Total Environ.* **940**, 173315 (2024).
- Jacobs, D. et al. Wastewater monitoring raises privacy and ethical considerations. *IEEE Trans. Technol. Soc.* **2**, 116–121 (2021).
- Sierra-Sosa, D., Telahun, M. & Elmaghraby, A. TensorFlow quantum: impacts of quantum state preparation on quantum machine learning performance. *IEEE Access* **8**, 215246–215255 (2020).

30. Schuld, M. & Patruccione, F. *Supervised Learning with Quantum Computers*. 139–167 (Springer Cham, 2018).
31. PennyLane. (n.d.). Quantum embedding [https://pennylane.ai/qml/glossary/quantum\\_embedding/](https://pennylane.ai/qml/glossary/quantum_embedding/).
32. Gundlapalli, P. & Lee, J. Deterministic and entanglement-efficient preparation of amplitude-encoded quantum registers. *Phys. Rev. Appl.* **18**, 024013 (2022).
33. Maheshwari, D., Sierra-Sosa, D. & Garcia-Zapirain, B. Variational quantum classifier for binary classification: real vs synthetic dataset. *IEEE Access* **10**, 3705–3715 (2022).
34. Nakaji, K. et al. Approximate amplitude encoding in shallow parameterized quantum circuits and its application to financial market indicators. *Phys. Rev. Res.* **4**, 023136 (2022).
35. Schuld, M., Bocharov, A., Svore, K. & Wiebe, N. Circuit-centric quantum classifiers. Preprint at <https://arxiv.org/abs/1804.00633> (2018).
36. Levine, Y., Yakira, D., Cohen, N. & Shashua, A. Deep learning and quantum entanglement: fundamental connections with implications to network design. Preprint at <https://arxiv.org/abs/1704.01552> (2017).
37. Deng, D.-L., Li, X. & Das Sarma, S. Quantum entanglement in neural network states. *Phys. Rev. X* **7**, 021021 (2017).
38. PennyLane. (n.d.). What is PennyLane <https://docs.pennylane.ai/en/stable/introduction/pennylane.html>.
39. Zhang, Y. et al. High dimensional missing data imputation for classification problems: a hybrid model based on K-nearest neighbor and genetic algorithm. In *Proc. 2022 International Symposium on Advances in Informatics, Electronics and Education (ISAIEE)* 572–578 (2022).
40. Abidin, N. Z. & Ismail, A. R. An improved K-nearest neighbour with grasshopper optimization algorithm for imputation of missing data. *Int. J. Adv. Intell. Inform.* **7**, 304–317 (2021).
41. Fadlil, A. K nearest neighbor imputation performance on missing value data graduate user satisfaction. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)* **6**, 570–576 (2022).
42. Moussa, C., Patel, Y. J., Dunjko, V., Bäck, T. & van Rijn, J. N. Hyperparameter importance and optimization of quantum neural networks across small datasets. *Mach. Learn.* **113**, 1941–1966 (2024).

## Acknowledgements

The authors would like to thank Thames Water Utilities Limited, Prof. Stephen Smith and Dr Jin Liu for the dataset support, their expert advice, and

comments on this work. The views expressed in the paper are those of the authors and do not necessarily represent the companies supporting the research.

## Author contributions

Y.M. and A.E. developed the models, analysed the data and wrote the manuscript. H.L. and A.S. contributed to the development and interpretation of the models and data analysis and revised the manuscript. P.L. contributed to the final version of the manuscript and supervised the project. All authors read and approved the final manuscript.

## Competing interests

The authors declare no competing interests.

## Additional information

**Supplementary information** The online version contains supplementary material available at <https://doi.org/10.1038/s41545-025-00440-y>.

**Correspondence** and requests for materials should be addressed to Po-Heng Lee.

**Reprints and permissions information** is available at <http://www.nature.com/reprints>

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© Crown 2025