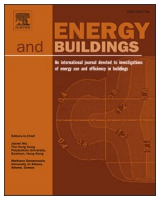
Energy & Buildings 346 (2025) 116156

Contents lists available at ScienceDirect

Energy & Buildings

journal homepage: www.elsevier.com/locate/enb



Can quantum computing surpass classical algorithms in optimizing building performance? A benchmark study with 15,000 simulations☆

Haidar Hosamo [a,*], Vagelis Plevris [b], Dimitrios Kraniotis [a], Christian Nordahl Rolfsen [a]

[a] Department of Built Environment, Oslo Metropolitan University, St. Olavs Plass, P.O. Box 4, NO-0130 Oslo, Norway
[b] Department of Civil and Environmental Engineering, College of Engineering, Qatar University, Doha P.O. Box 2713, Qatar

ARTICLE INFO

Keywords:
Quantum computing
Building performance optimization
Machine Learning
Non-dominated Sorting Genetic Algorithm II (NSGA-II)
Multi-Objective Particle Swarm Optimization (MOPSO)

ABSTRACT

Optimizing building performance is essential for enhancing energy efficiency and occupant comfort. This study evaluates the applicability of quantum computing–based optimization methods in the Architecture, Engineering, and Construction (AEC) industry by comparing the Quantum Approximate Optimization Algorithm (QAOA) and Quantum Annealing (QA) with classical multi-objective optimization algorithms, namely Non-dominated Sorting Genetic Algorithm II (NSGA-II) and Multi-Objective Particle Swarm Optimization (MOPSO). A dataset of 15,000 building simulations was used to train an Extreme Gradient Boosting (XGBoost) model for predicting total energy consumption (kWh/m^2/year) and Predicted Percentage of Dissatisfied (PPD) occupants. These predictions were then used in the optimization phase. NSGA-II produced the best trade-offs, achieving energy consumption between 17.84 and 19.84 kWh/m^2/year and PPD below 5.2 %, with strong diversity and convergence. QAOA executed faster (0.54 min) than NSGA-II (18.9 min) but resulted in higher energy values (31.85–55.62 kWh/m^2/year) and weaker solution quality. Quantum Annealing completed in 0.37 min but returned clustered solutions near 45.88 kWh/m^2/year. While the current limitations of quantum methods constrain their effectiveness, the findings indicate their potential as fast solvers in future building performance optimization workflows, particularly as hardware and algorithmic capabilities mature.

1. Introduction

In the pursuit of sustainable and energy-efficient building design, optimizing building performance is essential [1]. Modern buildings must minimize energy consumption, ensure occupant comfort, and maintain indoor air quality (IAQ), all within the constraints of complex, multivariable systems [2,3]. These requirements are often evaluated through metrics such as the Predicted Percentage of Dissatisfied (PPD), total energy consumption, and hours above critical temperature thresholds [4,5]. Addressing these challenges requires sophisticated optimization techniques capable of handling the intricate interdependencies among building parameters.

Conventional optimization methods, including genetic algorithms and particle swarm optimization, have been widely applied to address these challenges [6–8]. Genetic algorithms, inspired by principles of natural selection, evolve solutions over generations, making them effective for solving non-linear, multi-objective problems. However, the iterative nature of these methods often results in high computational costs, particularly for large-scale systems with extensive datasets. As buildings grow more complex and the demand for real-time optimization increases, the limitations of classical algorithms become more pronounced, necessitating the exploration of advanced computational approaches [9–11].

Machine learning, with its ability to analyze and extract patterns from large datasets, offers a robust solution [12]. Machine learning can identify critical interactions between building parameters, such as thermal inertia (mass), solar shading, ventilation rates, and indoor conditions (e.g., air temperature, PPD, CO_2 levels, lighting intensity etc.) which form the foundation for our subsequent optimization phase, enabling a targeted and computationally efficient approach to performance enhancement [13,14].

To further expand the scope of optimization capabilities, quantum computing introduces a new paradigm in optimization through its ability to process multiple variable combinations simultaneously

☆ This article is part of a special issue entitled: 'Decarbonising Built Env' published in Energy & Buildings.

* Corresponding author.
E-mail addresses: haidarho@oslomet.no (H. Hosamo), vplevris@qu.edu.qa (V. Plevris), dimitrios.kraniotis@oslomet.no (D. Kraniotis), christian.nordahl-rolfsen@oslomet.no (C.N. Rolfsen).

https://doi.org/10.1016/j.enbuild.2025.116156
Received 31 March 2025; Received in revised form 28 June 2025; Accepted 14 July 2025
Available online 15 July 2025

[15,16]. Quantum-inspired algorithms can deliver faster and more comprehensive optimization solutions [17]. Although quantum computing has shown promise in areas such as finance and logistics, its application in the Architecture, Engineering, and Construction (AEC) industry remains largely unexplored [18–21]. Quantum optimization has the potential to transform how building performance challenges are addressed, meeting the growing complexity of modern building systems.

This paper focuses on optimizing building performance through a combined approach that integrates machine learning with quantum-inspired algorithms, representing one of the first systematic applications of quantum computing in this domain. While previous studies have explored classical optimization and hybrid methods, this work advances the field by applying quantum algorithms—QAOA and Quantum Annealing—alongside machine learning to address the multi-objective trade-off between energy efficiency and thermal comfort. Machine learning techniques are used to uncover complex parameter interactions, such as those involving shading, ventilation rates, and indoor environmental quality, which then inform the optimization process. Unlike conventional approaches, this study benchmarks the resulting performance using thorough metrics supported by statistical validation tests. This framework provides novel evidence for the potential of quantum-inspired computing to enhance solution quality, convergence behavior, and real-time applicability in the AEC industry.

Following this introduction, **Section 2** reviews the applications of machine learning and optimization methods in building performance. **Section 3** explains the key dataset parameters, the machine learning techniques employed, and the configurations for both quantum-inspired and classical algorithms. **Section 4** evaluates the performance of these algorithms based on solution quality and computation time and discusses the practical implications of quantum algorithms in the AEC context. Finally, **Section 5** concludes with the potential of integrating machine learning and quantum computing to achieve sustainable building performance optimization.

2. Literature Review

2.1. Machine learning for building performance analysis

Machine learning has become fundamental for analyzing complex, high-dimensional building performance data, offering insights that enhance decision-making in energy management, thermal comfort, and indoor air quality [22]. Techniques such as regression models [23], decision trees [24], support vector machines [25], ensemble methods [26], and neural networks [27] are widely applied to model intricate relationships between building parameters, including shading, ventilation, and their impact on performance outcomes. Studies in building science show that machine learning models improve predictions for energy consumption, indoor temperature, and CO_2 levels, often uncovering insights that are challenging to derive through traditional statistical methods [28,29].

Each algorithm brings unique strengths to building analysis. For instance, decision trees and ensemble methods, such as random forests and gradient boosting, capture non-linear relationships between inputs, effectively handling the complex and interactive effects among parameters [30]. Neural networks, particularly deep learning models, excel in capturing subtle patterns within large, unstructured datasets, making them ideal for high-dimensional building data [31]. Support vector machines are effective in cases with smaller datasets and where classification tasks are involved, such as identifying energy-efficient versus inefficient scenarios [32]. These machine learning methods help identify critical variables that significantly impact energy efficiency and occupant comfort, establishing a strong basis for optimization efforts. Machine learning-driven analysis clarifies how building parameters interact under various conditions, supporting a refined approach to performance management.

2.2. Building performance optimization approaches

Optimization of building performance is necessary for achieving energy efficiency, occupant comfort, and environmental sustainability, with classical optimization algorithms widely applied in these efforts [33]. Among these algorithms, genetic algorithms, particle swarm optimization, Non-dominated Sorting Genetic Algorithm II (NSGA-II) [34,35], and simulated annealing have gained prominence for their ability to handle the multi-objective nature of building performance tasks. Genetic algorithms, for instance, optimize building parameters such as insulation thickness, HVAC setpoints, and window-to-wall ratios, improving energy efficiency while satisfying comfort constraints [2]. Particle swarm optimization has also been used to minimize HVAC energy use and regulate thermal comfort, adjusting variables like ventilation variables to meet energy-saving goals [36]. Simulated annealing, known for its robustness in exploring vast solution spaces, has found applications in optimizing the layout and design of building envelopes, contributing to improved thermal performance and indoor air quality [37].

When it comes to comfort, the PPD is a key metric for evaluating thermal comfort in indoor environments [5]. PPD represents the percentage of occupants likely to express dissatisfaction with thermal conditions. Ranging from 5 % (minimum dissatisfaction) to 100 %, PPD is derived from the Predicted Mean Vote (PMV) model, which integrates environmental and personal factors such as air temperature, metabolic rate, and clothing insulation [38]. Simulation tools like IDA Indoor Climate and Energy (IDA ICE) are often used to calculate PPD, offering detailed insights into how building performance affects occupant comfort under varying scenarios [39]. High PPD values can lead to increased energy use for thermal regulation, while overly strict comfort targets may result in unnecessary energy consumption [40]. Balancing PPD with energy efficiency and air quality is critical for sustainable building operations, ensuring comfort without excessive resource use [41]. Incorporating PPD into performance evaluations provides a comprehensive view of occupant-centric design and operational strategies, making it a cornerstone of sustainable and efficient building management [42].

However, these classical algorithms face limitations when tasked with handling the large-scale, complex interactions present in real-time building management. The high computational demands of processing vast combinations of design and operational parameters often lead to slower convergence rates and increased costs, particularly in multi-objective problems requiring rapid solutions [43]. For instance, optimization that involves balancing energy efficiency with air quality and thermal comfort can become computationally intensive, especially when constrained to real-time adaptability [44]. This complexity has led researchers to explore alternative approaches that address these limitations, among which quantum computing shows significant promise.

2.3. Quantum computing in optimization

Quantum computing introduces a fundamentally different approach to optimization, one that uses principles of quantum mechanics to process large, complex optimization tasks efficiently [45]. Quantum algorithms such as Quantum Approximate Optimization Algorithm (QAOA) [46], Variational Quantum Eigen solver (VQE) [47], and quantum annealing are capable of handling combinatorial optimization problems with a speed and accuracy beyond the reach of classical algorithms [48]. These algorithms exploit quantum phenomena like superposition and entanglement, allowing quantum computers to evaluate multiple solution states simultaneously [49]. This parallel processing capacity enables quantum computing to manage vast solution spaces in ways that traditional methods cannot, making it highly applicable to optimize scenarios where time efficiency and solution quality are essential.

The AEC industry has only recently started to explore quantum computing applications, and while research is limited, the findings

suggest that quantum computing could address the computational challenges associated with real-time building performance optimization [50–52]. In building science, where tasks often involve multi-objective optimization across energy, comfort, and environmental metrics, quantum computing's ability to efficiently explore and evaluate numerous parameter combinations could introduce a level of performance unattainable through classical methods.

2.4. Comparisons between quantum and classical algorithms

Comparative studies of quantum and classical optimization approaches in various fields have highlighted the unique advantages of quantum algorithms in computational speed, convergence stability, and solution quality [53,54]. In the context of multi-objective optimization, performance evaluation often relies on robust quality indicators such as hypervolume [55,56] and Inverted Generational Distance (IGD) [57,58], which assess both the diversity and proximity of generated Pareto fronts. However, only a few studies have employed these metrics in direct comparisons involving quantum algorithms. Furthermore, statistical methods such as the Friedman test [59] and the Nemenyi post-hoc test [60] are seldom applied in this domain, despite their importance in determining whether observed performance differences are statistically significant.

Within building performance optimization, studies comparing quantum and classical algorithms remain scarce, but preliminary research indicates that quantum algorithms could address the high computational demands associated with real-time building management. Classical optimization algorithms are often constrained by their sequential processing nature, which limits their efficiency in scenarios involving high-dimensional parameter spaces. Quantum computing's potential to evaluate multiple parameters simultaneously offers a pathway to overcoming these limitations [61]. Table 1 shows a comparative perspective underlines the research gap that this study addresses, as the AEC industry stands to benefit from demanding comparisons that assess quantum algorithms' feasibility in handling complex building performance challenges.

Out from that, the choice of QAOA and QA in this study was based on their suitability for combinatorial and high-dimensional optimization tasks, which are prevalent in energy and comfort trade-offs in building performance management. QAOA is particularly effective for problems expressible in Quadratic Unconstrained Binary Optimization (QUBO) form, a common formulation in building optimization tasks where discrete decision variables are involved. QA, on the other hand, is designed to search for low-energy configurations in complex landscapes, making it well-suited for energy minimization tasks in multi-objective settings. These algorithms also align well with current quantum simulation platforms and are among the most mature and accessible quantum optimization approaches available today. Other quantum methods like Variational Quantum Eigensolvers (VQE) [47] or Grover's search [73] were excluded due to their unsuitability for multi-objective combinatorial optimization and lack of integration with QUBO-based frameworks used in this study. Moreover, hybrid variants [74] were deliberately omitted to isolate and benchmark the pure quantum components before integrating them into hybrid pipelines in future work.

2.5. Research gap and contribution

Despite the increasing use of classical optimization and machine learning in building performance studies, existing approaches often fall short in managing the real-time, multi-objective nature of building operations. Prior research has mainly addressed either energy use or thermal comfort in isolation or applied classical algorithms that become computationally inefficient when both objectives must be optimized simultaneously in high-dimensional design spaces.

This study introduces a novel hybrid framework that combines a surrogate model built with XGBoost and quantum optimization techniques, Quantum Approximate Optimization Algorithm (QAOA) and Quantum Annealing. Unlike prior studies, this work focuses explicitly on the joint optimization of two competing objectives: energy consumption (kWh/m^2/year) and thermal comfort (measured by Predicted Percentage of Dissatisfied occupants, PPD). It is, to our knowledge, the first study to benchmark quantum and classical multi-objective optimization algorithms (NSGA-II and MOPSO) using an identical building simulation dataset of 15,000 samples for a fair and controlled comparison.

Table 1
Comparison of classical and quantum algorithms for optimization in building management.

Aspect	Classical Algorithms	Quantum Algorithms	References
Computational Speed	Sequential processing limits scalability in high-dimensional problems.	Parallel processing allows for faster exploration of solution spaces, improving computation speed.	[62,63]
Optimization Strategy	Often relies on gradient-based or heuristic techniques which may not fully explore the solution space.	Quantum algorithms, like QAOA, use non-adiabatic mechanisms to bypass limitations of traditional methods.	[64,65]
Handling Complex Constraints	Limited capacity for multi-objective and high-constraint optimization due to computational bottlenecks.	Efficient in handling complex constraints, with hybrid approaches to improve solution quality.	[66,67]
Algorithmic Flexibility	Limited by fixed heuristics and sequential methods in dynamic environments like building management.	Quantum algorithms offer flexible parameter tuning and hybrid classical quantum approaches to optimize adaptability.	[68,69]
Scalability with Data Size	Slower as data size and complexity increase due to sequential processing limitations.	Quantum computing shows scalability in large datasets with high-dimensional parameter spaces.	[19,70]
Application for Real-time Management	Limited in real-time adjustments, often not viable for dynamic building optimization tasks.	Quantum systems can potentially meet real-time demands through rapid parallel computations.	[71]
Energy Efficiency in Computation	High energy consumption due to extended computation times in complex optimization.	Potential for reduced energy use in computation due to efficient parallel processing.	[72]

The main contributions of this work are:

1. Simultaneous optimization of energy and comfort using both classical and quantum techniques.
2. Evaluation of optimization performance using hypervolume and IGD metrics for convergence and diversity analysis.
3. Validation of machine learning–optimized solutions through simulation to assess surrogate model accuracy.
4. Statistical analysis with Friedman and Nemenyi tests to confirm significant performance differences.
5. Demonstration of the practical potential and limitations of quantum algorithms in the built environment.

This work advances the field by introducing a scalable, reproducible,

and computationally efficient framework for energy-comfort optimization in buildings, offering a new direction for real-time control strategies using quantum computing.

3. Methodology

A structured methodology is applied to optimize building energy performance, integrating simulation, machine learning, and optimization techniques, as shown in Fig. 1. The following subsections provide a detailed explanation of each step in the process.

3.1. Case study

The dataset used in this study was developed based on Tvederstrand upper secondary school (Fig. 2) that adheres to the Norwegian Technical Regulation under the Planning and Building Act (TEK10) [75] and the Norwegian Standard NS3701 [76]. These regulations set stringent criteria for passive houses and low-energy commercial buildings, ensuring high energy efficiency, occupant comfort, and reduced environmental impact. The design of the baseline building incorporates several advanced features that contribute to its performance. For instance, vertical fins installed on the windows provide effective shading, significantly reducing solar heat gains. The mechanical balanced ventilation system operates with a rotary heat recovery efficiency of 85 percent, minimizing energy losses during ventilation and ensuring a well-regulated indoor climate.

The building's baseline parameters were carefully selected to reflect realistic and energy-efficient configurations under Norwegian climatic conditions. Key thermal characteristics include U-values for external walls, roof, windows, and ground floor, which are critical in determining the overall thermal performance of the building envelope. Airtightness is maintained at a high standard (0.35 1/h), limiting heat loss through infiltration. The window-to-wall ratio (WWR) is set at 40 percent, balancing daylight penetration with thermal insulation. Reflectance and shading factor values further optimize solar heat gain and daylight distribution.

Additional operational features include centralized heating and cooling systems with efficiencies of 90 percent, designed to maintain setpoint temperatures of 21 °C for heating and 24 °C for cooling. The ventilation system is equipped with specific fan power (SFP) of 1.4 kW/(l/s) and airflow rates tailored to occupancy patterns, ensuring adequate ventilation during both occupied (2.48 l/(m^2·s)) and unoccupied (0.81 l/(m^2·s)) periods. Lighting power density is set at 5 W/m^2, contributing to energy efficiency while maintaining visual comfort. Internal heat gains from occupancy patterns and energy loads were modelled in compliance with the NS3031 standard, a widely referenced guideline for simulating realistic building performance under Norwegian conditions. The initial parameters used in the baseline simulation are presented in Table 2. These parameters form the foundation for subsequent analyses and simulations, offering a detailed insight into the building's design and operational characteristics.

All experiments in this study were conducted using Python 3.13. Machine learning models were implemented using widely adopted open-source libraries. Specifically, XGBoost was used to construct surrogate models predicting energy consumption and comfort metrics, while Scikit learn was used to implement Support Vector Regression, Artificial Neural Networks, and K Nearest Neighbors. For data preprocessing, analysis, and visualization, NumPy, Pandas, and Matplotlib were employed. The classical optimization algorithms NSGA II and MOPSO were executed using the Platypus framework, which supports multi-objective evolutionary algorithms. Quantum Approximate Optimization Algorithm was simulated using PennyLane in combination with Qiskit, while Quantum Annealing was simulated locally using D-Wave's Ocean SDK with the dimod backend, due to lack of access to real quantum hardware. All computations were developed and executed on a high-performance gaming laptop running Windows 11, equipped with an Intel Core Ultra 7 155H processor, 32 GB RAM, and an NVIDIA RTX 4060 GPU. Further implementation details are provided in Section 4.5.2.

3.2. Simulation framework and dataset characteristics

The dataset consists of 15,000 simulation runs systematically generated through pairwise testing [2] and IDA ICE to explore the interactions between input parameters and their influence on energy performance. This marks an extension of our previous work [77], which included 12,000 runs, with an additional 3,000 simulations added to increase complexity and enhance the robustness of the comparison between quantum and classical optimization algorithms.

Pairwise testing enabled a structured and efficient approach to generate a comprehensive set of simulations while minimizing redundancy. High-energy-consuming simulations identified in previous work were excluded, and the dataset was updated with new simulations, ensuring that the maximum energy consumption does not exceed 85 kWh/m^2/year.

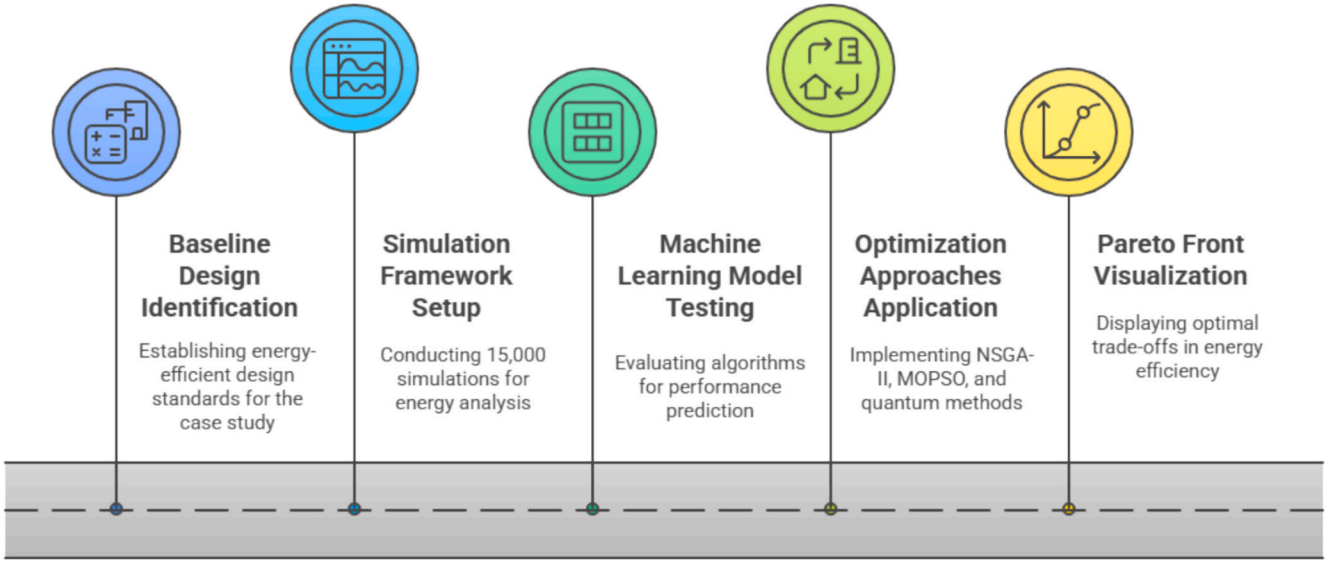


Fig. 1. Overview of the methodology used in this study.

Fig. 2. Illustration of the Tvederstrand upper secondary school.

Table 2
Initial parameter values for building performance simulation.

Parameter	Initial Value
External wall U-value (W/m^{2}K)	0.15
Roof U-value (W/m^{2}K)	0.11
Window U-value (W/m^{2}K)	0.8
Ground floor U-value (W/m^{2}K)	0.06
Thermal bridge U-value (W/m^{2}K)	0.03
Airtightness (1/h)	0.35
WWR	40 %
g-value	0.2
Reflectance	0.55
Ventilation system	Mechanical balanced with rotary heat recovery (85 % efficiency)
Specific Fan Power (kW/(l/s))	1.4
Mechanical ventilation airflow (l/(m^2·s))	2.48 (occupied), 0.81 (unoccupied)
Room heating system	Centralized heating (90 % efficiency)
Room cooling system	Centralized water cooling for AHU supply air
Heating setpoint (°C)	21
Cooling setpoint (°C)	24
Lighting power (W/m^2)	5
Occupancy (W/person)	108
Energy consumption (kWh/m^2/year)	61.07

Key parameters, such as U-values of building components, WWR, shading factors, and ventilation rates, were systematically varied to evaluate their impact on total energy consumption and other performance metrics. The simulations were designed to capture realistic variations in performance across a wide range of design configurations, providing a strong foundation for the optimization process. For instance, the interaction between shading devices and g-values was examined to quantify their combined effect on solar heat gain and cooling demand. Similarly, ventilation rates and airtightness levels were adjusted to evaluate their influence on energy efficiency, thermal comfort and IAQ.

3.3. Visualization and statistical summary

The histogram in Fig. 3 illustrates the frequency distribution of total energy consumption (kWh/m^2/year) across the simulated dataset. The data exhibit a bell-shaped distribution, with most values concentrated around the mean of 44.481 kWh/m^2/year, as reflected in the statistical summary in Table 3. The minimum and maximum values for total energy are 8.636 (causing high PPD) and 83.491 kWh/m^2/year, respectively, demonstrating the variability within the dataset.

Key parameters influencing energy consumption are also summarized in Table 3, which provides insights into the dataset's characteristics. For instance, Room depth ranges from 4 to 8 m, with a mean of 6.005 m, while the U-value of the façade varies between 0.12 and 0.2 W/(m^2·K), averaging 0.16. Similarly, the WWR spans from 30 % to 90 %, with a mean of 59.946 %, and shading factors range between 0.2 and 0.8. The dataset's high PPD, with a mean of 62.897 %, reflects the significance of balancing design parameters to optimize performance and comfort.

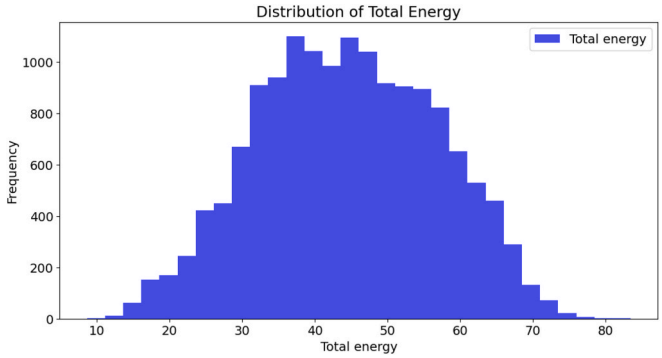


Fig. 3. Frequency distribution of total energy consumption (kWh/m^2/year) across the simulated dataset.

3.4. Machine learning for predictive modelling

Five widely used and well-established machine learning models were employed to analyze building performance data, aiming to identify the best-performing model as input for optimization algorithms. The selected models are Extreme Gradient Boosting (XGBoost), Random Forest (RF), Support Vector Regression (SVR), Neural Network (NN), and K-Nearest Neighbors (KNN). These algorithms were chosen for their effectiveness in capturing linear and non-linear relationships, as well as their adaptability to diverse data structures.

The best-performing model will be identified based on its predictive accuracy, robustness, and ability to generalize across unseen data. Performance metrics such as Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and the Coefficient of Determination (R^2) will be used to evaluate the models during cross-validation and testing phases. The model with the lowest RMSE and highest R^2 will be selected for its ability to produce reliable predictions of performance metrics such as energy consumption and PPD.

This selected model is then integrated into the optimization phase, where it serves as a predictive tool for evaluating and guiding parameter adjustments. Its role is critical for enabling the optimization algorithms to focus on high-impact variables and achieve better energy efficiency, thermal comfort, and overall building performance.

3.4.1. Model descriptions

- **Extreme Gradient Boosting (XGBoost) [78]:** An advanced boosting algorithm that incorporates regularization to prevent overfitting. Its objective function combines a loss term and regularization:

$$L(\Theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{j=1}^{T} \Omega(T_j) \quad (1)$$

Table 3
Statistical summary of key building parameters and performance metrics.

	mean	std	min	25 %	50 %	75 %	max
Room depth (m)	6.005	1.149	4	5.008	6.01	7	8
U-val, façade W/(m^2·K)	0.16	0.023	0.12	0.14	0.16	0.18	0.2
WWR (%)	59.946	17.331	30	44.982	59.904	74.87	90
g-value	0.441	0.11	0.25	0.35	0.44	0.537	0.63
Overhang (cm)	29.935	17.268	0	15	30	45	60
Shading factor	0.502	0.174	0.2	0.35	0.5	0.652	0.8
Shade on (lx)	68826.13	17928.66	38,003	53405.5	68,860	84078.58	99,987
Ventilation rate (ACH)	1.249	0.431	0.5	0.88	1.242	1.62	2
Internal load w/ m^2	17.514	7.23	5	11.21	17.53	23.84	30
Lighting w/ m^2	4.983	1.75	2	3.501	5	6.381	8
Reflectance	0.589	0.11	0.4	0.491	0.59	0.684	0.78
PPD (%)	62.897	31.108	5.013	35.497	65.801	95.888	100
Total energy (kWh/ m^2./year)	44.481	12.367	8.636	35.39	44.404	53.904	83.491

where l represents the loss function, Ω is the regularization term, and T_j represents individual trees.

- **K-Nearest Neighbors (KNN)** [79]**:** A non-parametric algorithm that predicts output based on the k-nearest data points in feature space. The prediction for regression is:

$$\hat{y} = \frac{1}{k}\sum_{i=1}^{k} y_i \tag{2}$$

where y_i are the outputs of the k-nearest neighbors.

- **Random Forest (RF)** [80]**:** An ensemble method that builds multiple decision trees and averages their predictions to reduce overfitting and improve generalization. For regression:

$$\hat{y} = \frac{1}{N}\sum_{i=1}^{N} T_i(x) \tag{3}$$

This equation calculates the average prediction $\hat{y}$ across N trees $T_i(x)$.

- **Support Vector Regression (SVR)** [81]**:**

It is a predictive algorithm designed to estimate continuous outputs by constructing a model that balances prediction accuracy and model complexity. The SVR model finds a hyperplane that predicts the target values while allowing a margin of tolerance (∈) around the observed data. Predictions are made using the learned weights (w) and bias (b), with the optimization process determining these parameters. The prediction for a new input x_i is given by:

$$\hat{y} = \langle w, x_i \rangle + b \tag{4}$$

The training process minimizes model complexity and ensures predictions stay within the ∈-margin of the true values, formalized as:

$$\text{Minimize } \frac{1}{2}||w||^2 \text{ subject to } |y_i - (\langle w, x_i \rangle + b)| \leq \epsilon \tag{5}$$

- **Artificial Neural Network (ANN)** [82]**:**

An Artificial Neural Network is a predictive model inspired by the structure and functioning of the human brain. It consists of interconnected layers of neurons that transform inputs into outputs through weighted sums and activation functions. The ANN learns patterns in the data by minimizing an objective function during training. For regression tasks, the Mean Squared Error (MSE) loss function is commonly used:

$$\mathscr{L} = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{6}$$

Here:

- y_i is the true output,
- $\hat{y}_i$ is the predicted output,
- n is the number of data points.

Once trained, the network predicts the output ($\hat{y}_i$) for a new input (x) by passing it through multiple layers of weighted transformations and non-linear activation functions, mapping input features to the desired output. ANNs are well-suited for capturing complex, non-linear relationships in data, making them highly versatile for a variety of predictive tasks.

3.4.2. Training and testing process

The dataset was divided into training (80 %), validating (10 %) and testing (10 %) subsets. K-fold cross-validation (k = 5) was employed on the training data to ensure robustness. For each model, hyperparameters were fine-tuned using grid search, optimizing performance on validation data.

3.4.3. Performance metrics

The models were evaluated using:

- **Root Mean Squared Error (RMSE):**

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{7}$$

- **Coefficient of Determination (R^2):**

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \tag{8}$$

- **Mean Absolute Error (MAE):**

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{9}$$

3.4.4. Fine-Tuned hyperparameters

The hyperparameters of each machine learning model were fine-tuned to achieve optimal predictive performance. Hyperparameters are important settings that control the model's behavior and structure, influencing its ability to learn from the data [83]. For example, parameters like learning rate in XGBoost govern the step size during weight updates, while max_depth defines the maximum layers or splits in tree-based models, controlling their capacity to capture complexity. In KNN, n_neighbors determines how many closest data points influence the prediction, and weights specify whether closer points carry more importance. The kernel parameter in SVR defines the transformation of input data into higher-dimensional spaces to capture non-linear relationships, while C balances model complexity and accuracy. Neural networks rely on parameters such as input_layer and hidden_layers, which define the number of neurons in the input and intermediate

layers, respectively, enabling the model to process input features and learn intricate patterns. The activation function, such as ReLU, introduces non-linearity to neural networks, making them capable of learning complex relationships.

These hyperparameters were tuned through grid search to ensure the models achieved high accuracy and generalizability, as summarized in Table 4. For XGBoost, 45 configurations were evaluated by combining learning rate values (0.01, 0.05, 0.1, 0.2, 0.3), max_depth (3, 4, 5, 6, 7), and n_estimators (50, 100, 150). K-Nearest Neighbors was tested across 12 settings using n_neighbors values (3, 5, 7, 9) and weights ('uniform', 'distance'), with distance calculated using the Minkowski metric (p = 2). Random Forest was evaluated over 27 configurations by varying n_estimators (100, 150, 200), max_depth (10, 15, 20), and min_samples_split (2, 5, 10). For Support Vector Regression, 16 parameter combinations were tested using the RBF kernel, across C values (1, 10, 100, 1000) and epsilon (0.01, 0.1, 1, 10). The neural network model was explored using 18 different configurations by adjusting the number and size of hidden layers [(16,), (16, 32), (16, 32, 16)] and activation functions (ReLU, Tanh). The best parameters were summarized in Table 4.

The visualization in Fig. 4 illustrates the structure of a single decision tree within the XGBoost model, trained on building performance data. Each orange box represents a decision node, with conditions that split the dataset based on feature thresholds, such as $f3 < 0.277552515$ (where $f3$ is one of the input features). Blue arrows denote paths taken when the condition is false, while red arrows represent paths when the condition is true. The turquoise ellipses at the ends of the branches are leaf nodes, which provide the final predicted values for that path. The structure demonstrates how XGBoost sequentially narrows the decision paths to optimize the prediction of building performance outcomes. This hierarchical representation shows the decision-making process, emphasizing how individual features influence the final predictions. Additionally, Fig. 5 presents the structure of the neural network trained on the same dataset. It consists of an input layer with 10 neurons representing building performance parameters, followed by three hidden layers with 16, 32, and 16 neurons, respectively, and an output layer with 2 neurons predicting Total Energy and Predicted PPD. Connections between neurons illustrate how input features are transformed through weighted computations and activation functions across layers.

The outputs from the best-performing model will be used as inputs for the optimization phase, enabling the algorithms to focus on high-impact variables. This streamlined the process of achieving precise and computationally efficient solutions for sustainable building management.

3.5. Optimization algorithms

3.5.1. Classical optimization for Multi-Objective optimization

Classical optimization algorithms, specifically NSGA-II [34] and MOPSO [84], were employed to minimize energy consumption (E) and PP D in building performance. These algorithms relied on a fitness function derived from machine learning predictions, which accurately captured the complex, non-linear relationships between building parameters and performance metrics.

Table 4
Best hyperparameters for each model after fine-tuning.

Model	Hyperparameters
XGBoost	Learning rate = 0.1, max depth = 4, n estimators = 50, visualization includes split thresholds and leaf predictions for PPD and energy.
K-Nearest Neighbors	N neighbors = 5, weights='distance', p = 2
Random Forest	N estimators = 200, max depth = 15, min samples split = 5
Support Vector Regression	kernel = Radial Basis Function (RBF), C = 10, epsilon = 0.1
Neural Network	Input layer = 10 neurons, hidden layers = (16, 32, 16), output layer = 2 neurons, activation = ReLU (Rectified Linear Unit)

The optimization was guided by the following multi-objective formulation:

$$\text{Minimize } F(x) = [E(x), \text{PPD}(x)] \tag{10}$$

where $E(x)$ and $\text{PPD}(x)$ were predicted by the trained machine learning models. The algorithms aimed to identify Pareto-optimal solutions, where no single objective could be improved without worsening the other. This approach ensured that the resulting solutions represented trade-offs between energy efficiency and occupant comfort.

NSGA-II, a widely used evolutionary algorithm, was designed to find Pareto-optimal solutions by evolving a population of candidate configurations [35], as can be seen in Fig. 6 (a). It began with a random initialization of solutions and ranked them using Pareto dominance, where a solution was considered optimal if no other solution performed better in both objectives. To ensure diversity along the Pareto Front, NSGA-II employed a crowding distance metric, which prevented solutions from clustering and maintained a well-distributed set of trade-offs. Genetic operators such as selection, crossover, and mutation were applied iteratively to improve the population. For this study, the parameters were tuned as follows: a population size of 100, 200 generations, a crossover probability of 0.8, and a mutation probability of 0.05. These settings allowed the algorithm to balance exploration of the solution space with convergence toward the Pareto Front.

MOPSO, an extension of the Particle Swarm Optimization algorithm, used a swarm of particles to explore the solution space (Fig. 6 (b)). Each particle represented a potential configuration, and its movement was guided by the predicted fitness values of energy consumption and discomfort. MOPSO maintained a repository of non-dominated solutions to approximate the Pareto Front, ensuring that the diversity and quality of solutions were preserved. Particle velocities and positions were updated using the following equations:

$$v_i(t+1) = \omega v_i(t) + c_1 r_1 (p_i - x_i(t)) + c_2 r_2 (g_{\text{leader}} - x_i(t)) \tag{11}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{12}$$

where the $v_i(t)$ and $x_i(t)$ vectors denote the velocity and position of particle i at iteration t, ω is the inertia weight controlling exploration and exploitation, c_1 and c_2 are cognitive and social coefficients, and p_i and g_{leader} are the particle's personal best and the swarm's global best positions, respectively. The random numbers r_1 and r_2 add stochasticity to the search process. The parameters for MOPSO included a swarm size of 100, a maximum of 200 iterations, an inertia weight of 0.7, and cognitive and social coefficients set to 1.5 each.

3.5.2. Quantum-Inspired optimization

Quantum-inspired optimization techniques, such as the Quantum Approximate Optimization Algorithm (QAOA) [64] and quantum annealing [85], provide classical quantum simulators to emulate the behavior of quantum systems. These simulators enable researchers to test and evaluate the benefits of quantum-inspired algorithms without requiring access to physical quantum computers, providing a bridge between classical and quantum computational paradigms.

The implementation of QAOA begins with the encoding of the optimization problem as a cost function derived from a machine learning model. This cost function combines energy consumption (E) and the PPD and is expressed as

$$C(\boldsymbol{z}) = w_1 E(\boldsymbol{z}) + w_2 \text{PPD}(\boldsymbol{z}) \tag{13}$$

Where $\boldsymbol{z}$ represents the decision variables.

The quantum circuit for QAOA (Fig. 7) is designed by representing the cost function as a quantum operator acting on qubits. Initial state preparation involves initializing all qubits in a uniform superposition using Hadamard gates. The cost function is then encoded into a diagonal operator acting on the qubits, followed by the application of mixing

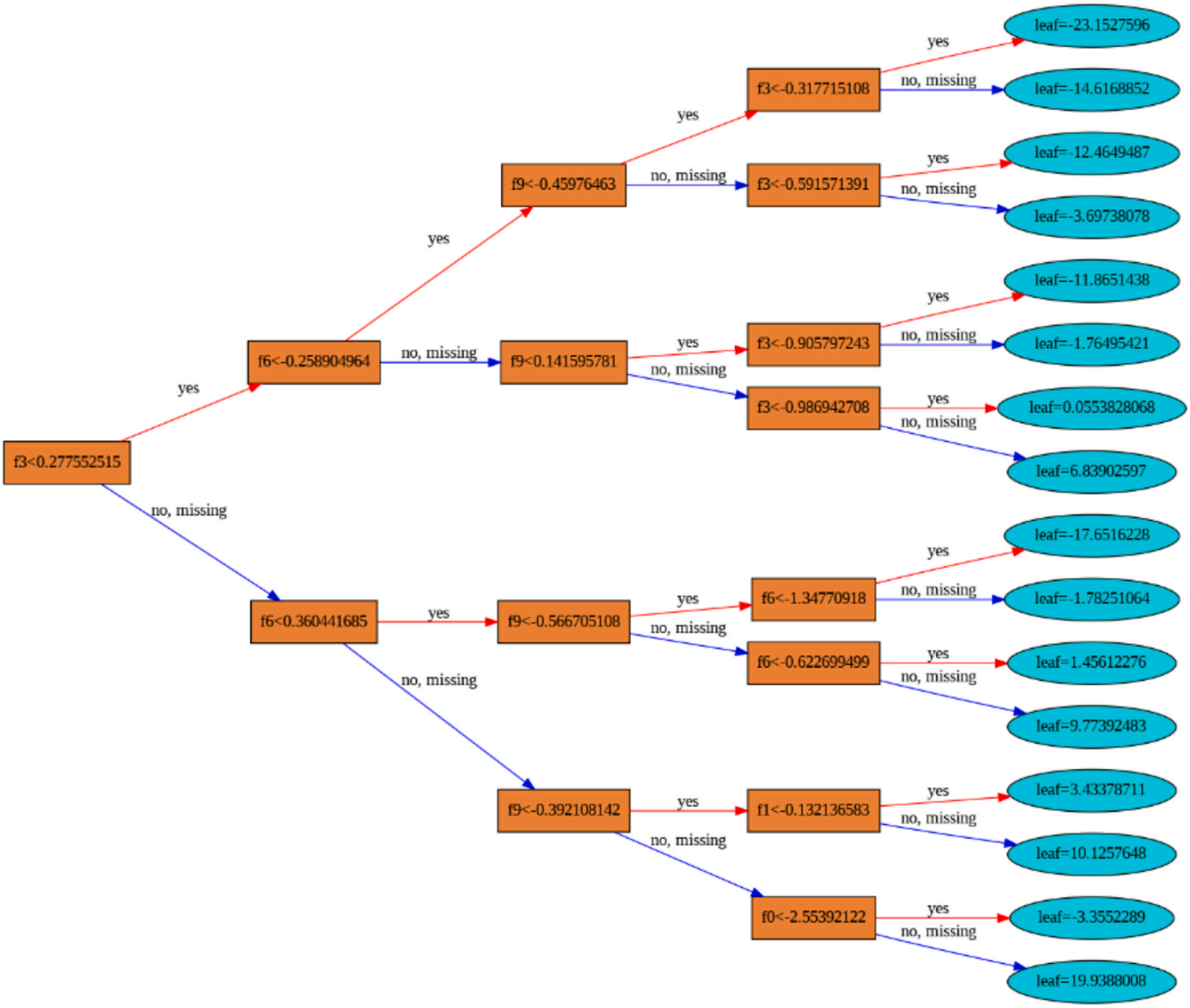


Fig. 4. XGBoost tree structure illustrating decision nodes, feature thresholds, and final predictions at leaf nodes for the building performance dataset.

operators using X-rotations to explore possible solutions.

Parameter optimization for QAOA is achieved through classical optimization techniques, such as gradient descent. The quantum circuit parameters γ and β are iteratively updated using the formula

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla \mathscr{C}(\theta) \tag{14}$$

η is the learning rate, and $\nabla \mathscr{C}(\theta)$ represents the gradient of the cost function. The algorithm was simulated using PennyLane, a Python-based library for hybrid quantum–classical computation. PennyLane interfaced with the NumPy backend to simulate quantum gates and measure outcomes, ensuring compatibility with classical optimization routines and enabling precise tracking of convergence. Execution parameters included a circuit depth of 3, 10 qubits, a gradient descent optimizer, and a learning rate of 0.01 over 100 iterations.

Quantum annealing (Fig. 7), another quantum-inspired optimization technique, maps the machine learning-derived cost function to the energy states of a quantum system. Each qubit represents a decision variable, and the interactions between qubits encode relationships within the cost function. The Hamiltonian of the system that models the problem can be represented as

$$H = \sum_i w_1 E_i z_i + w_2 \mathrm{PPD}_i z_i + \sum_j J_{ij} z_i z_j \tag{15}$$

Here, z_i are binary variables representing optimization decisions, and J_{ij} encodes pairwise interactions. Quantum annealing is simulated by gradually transitioning the system from an initial Hamiltonian, which represents a uniform superposition, to a final Hamiltonian that encodes the optimization problem. This process follows the time-dependent Hamiltonian, which is the annealing schedule, and T is the total annealing time.

$$H(t) = (1 - s(t))H_{\mathrm{init}} + s(t)H_{\mathrm{final}} \tag{16}$$

Simulations of quantum annealing were carried out using the D-Wave Ocean SDK [86], which provided a classical simulation of the annealing process. The optimization problem was embedded onto a Chimera graph topology to ensure compatibility with the annealing simulation. Execution parameters included 10 qubits, a total annealing time of 100 ms, and a linear annealing schedule.

Equations (17) to (21) detail the mathematical formulations underpinning quantum-inspired optimization techniques. Equation (17) represents the cost function C_z, combining energy consumption (E) and the

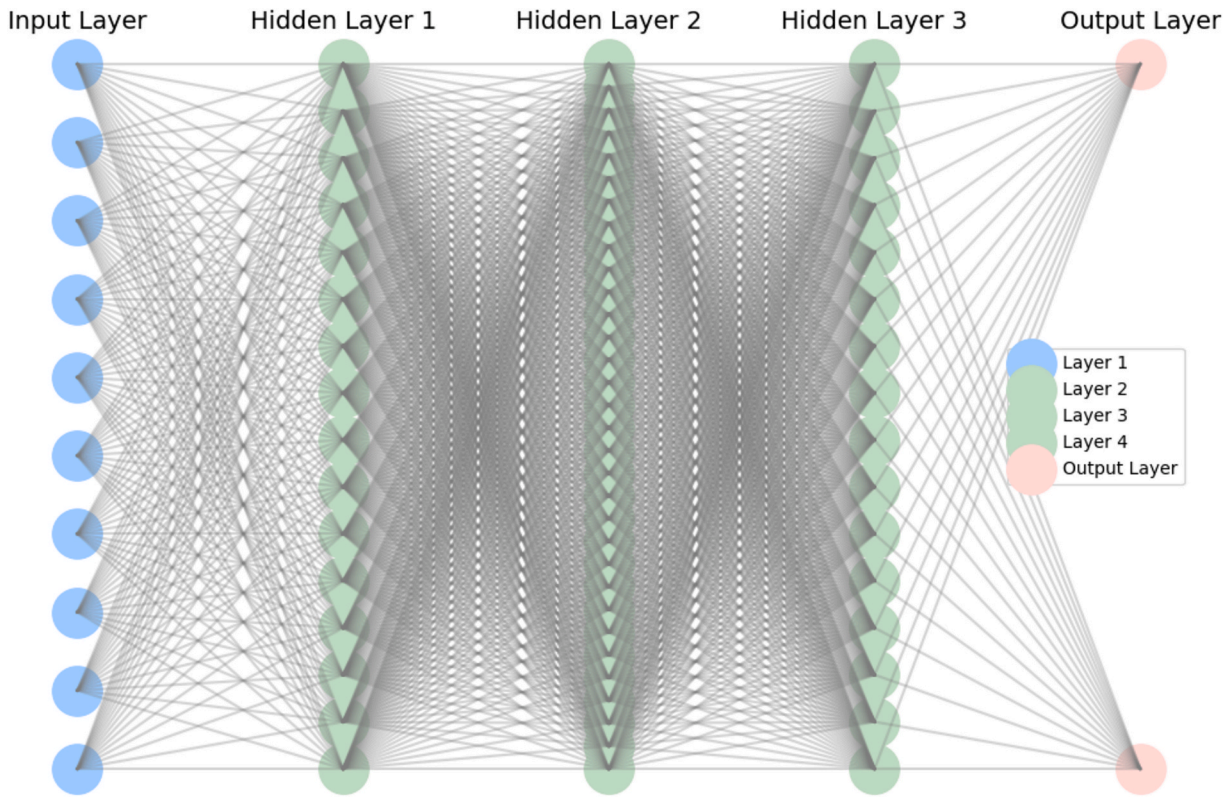


Fig. 5. Neural network architecture used for building performance prediction.

PPD with weights $w_1 and w_2$ reflecting their relative importance. Equation (18) outlines the gradient descent rule for parameter updates in QAOA, iteratively optimizing the circuit parameters (θ) to minimize the cost function. Equation (19) describes the Hamiltonian formulation in quantum annealing, encoding the optimization problem into energy states where zi are decision variables, and J_{ij} captures pairwise interactions. Equation (20) introduces the time-dependent Hamiltonian, transitioning the system from an initial uniform superposition to the problem-specific Hamiltonian through the annealing schedule defined in Equation (21) ensures a gradual and controlled evolution over the total annealing time (T). Together, these equations form the foundation for implementing and simulating quantum-inspired optimization approaches.

$$C(z) = w_1 E(z) + w_2 \mathrm{PPD}(z) \tag{17}$$

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla \mathscr{C}(\theta) \tag{18}$$

$$H = \sum_i w_1 E_i z_i + w_2 \mathrm{PPD}_i z_i + \sum_j J_{ij} z_i z_j \tag{19}$$

$$H(t) = (1 - s(t))H_{\mathrm{init}} + s(t) H_{\mathrm{final}} \tag{20}$$

$$s(t) = \frac{t}{T} \tag{21}$$

4. Results and discussion

4.1. Machine learning model performance

To identify the most suitable predictive model for building performance optimization, five widely used machine learning algorithms were tested: Random Forest, SVR, KNN, Neural Network, and XGBoost. These models were evaluated based on their ability to predict total energy consumption (kWh/m^2/year) and the PPD occupants. The evaluation metrics included RMSE, MAE, and R^2.

Table 5 presents a comparative analysis of the machine learning models. The results demonstrate that XGBoost and Random Forest exhibited the highest predictive accuracy across both performance metrics. XGBoost achieved an energy RMSE of 1.89 kWh/m^2/year and an R^2 of 0.99, while Random Forest yielded similar results with an energy RMSE of 1.93 kWh/m^2/year and an R^2 of 0.98. Both models also performed well in predicting PPD, with Random Forest achieving an R^2 of 1.00 and XGBoost attaining 1.00. In contrast, SVR, KNN, and Neural Network models displayed higher errors and lower predictive accuracy, suggesting their limitations in capturing the intricate relationships between building parameters. Table 6 presents a selection of optimal design configurations identified through NSGA-II that achieve a PPD below 10%.

The actual vs. predicted plots for the five models are illustrated in Fig. 8, where the diagonal red dashed line represents an ideal fit. It is evident that XGBoost and Random Forest closely align with the actual values, demonstrating strong predictive capabilities. Conversely, SVR, KNN, and Neural Network models exhibit significant deviations, particularly in energy predictions, indicating challenges in generalizing across unseen data.

To further justify the selection of XGBoost over other models, SHAP (SHapley Additive exPlanations) analysis [87] analysis was performed on the XGBoost model. was applied to uncover the internal decision logic of the model and offer transparency in feature influence. Unlike traditional feature importance metrics, SHAP provides consistent and locally accurate explanations by quantifying how each feature value contributes to individual predictions.

As demonstrated in Fig. 9 and Fig. 10, SHAP values enabled a comparative understanding of the relative impact of design parameters on both total energy and thermal comfort. XGBoost's compatibility with tree-based SHAP methods, its ability to handle non-linearity, feature interactions, and regularization were key reasons for its superior predictive performance and interpretability. Models such as neural networks or SVR may have comparable accuracy in certain conditions, but

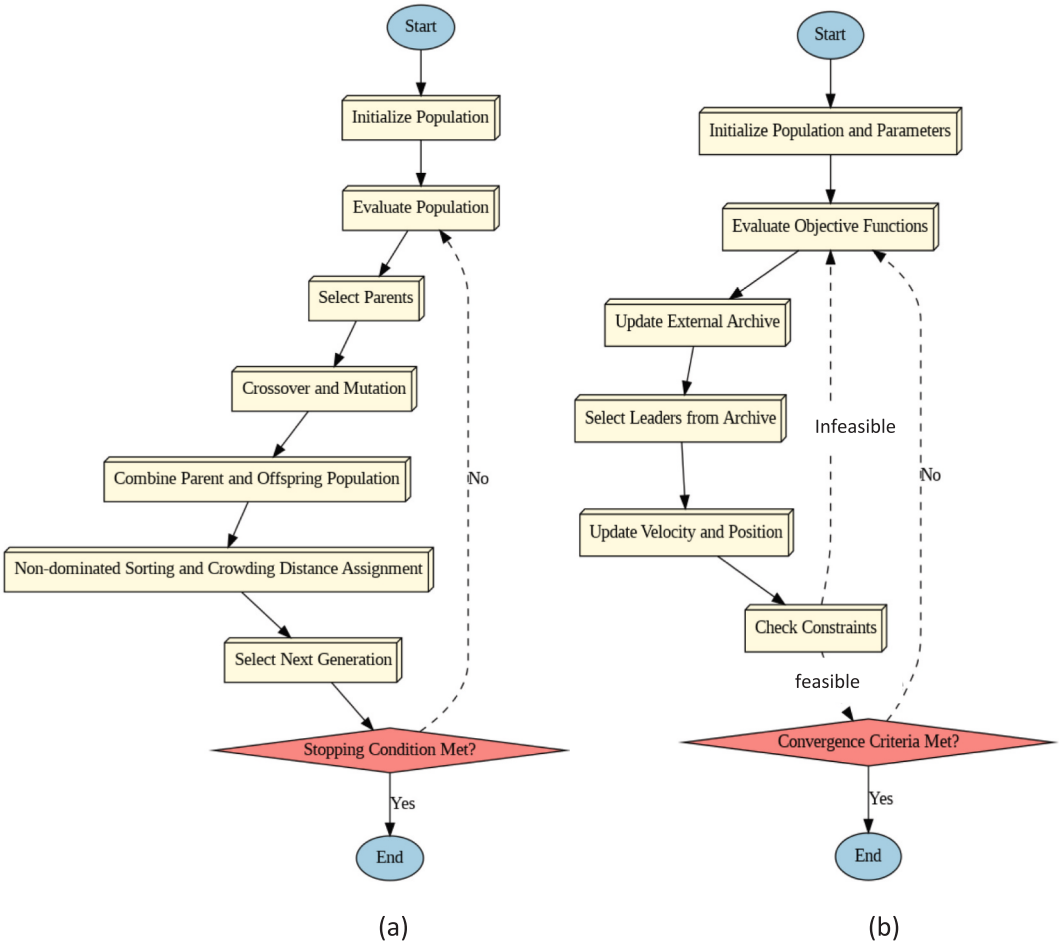


Fig. 6. Optimization Techniques (a) Flowchart of the NSGA-II Algorithm where the process begins with initializing a population, followed by evaluation, selection, crossover, mutation, and combination of populations. (b) Flowchart of the MOPSO Algorithm where the algorithm starts with the initialization of population and parameters, followed by objective function evaluation.

they lack the transparent architecture and direct SHAP integration that XGBoost provides. Thus, the choice of XGBoost is validated by its performance metrics and by its suitability for interpretable decision-making in building performance optimization.

Fig. 9 shows that ventilation rate (ACH) had the highest impact, followed by internal load, room depth, and window-to-wall ratio (WWR). High ventilation rates and internal loads were associated with increased energy demand, which aligns with thermodynamic principles. Room depth and WWR influenced solar gains and heat distribution, explaining their significance. In contrast, U-value of the façade had a surprisingly minor SHAP impact despite its well-known relevance in heat loss. This may be attributed to a limited range of variation in the input data or its secondary role compared to dynamic parameters like ventilation and internal loads in this simulation context. Similarly, shading factor and g-value showed lower influence, possibly due to compensatory interactions among solar control features.

Fig. 10 presents the SHAP summary for PPD prediction. Internal load was the most dominant factor, followed by overhang, reflectance, and shading control variables. This pattern highlights that occupant comfort is more sensitive to internal gains and shading strategies than to envelope parameters like U-value or WWR. The contrast between the two plots emphasizes the multi-objective nature of optimizing both energy and comfort.

4.2. Pareto front comparison of NSGA-II, MOPSO, and QAOA

Following the selection of XGBoost as the best-performing model, its predictions were used as input fitness function for the optimization phase. To assess the optimization performance of classical and quantum-inspired methods, a Pareto front comparison was conducted between NSGA-II, MOPSO, and QAOA. The multi-objective optimization problem aimed to minimize both total energy consumption (kWh/m^2/year) and the PPD.

Fig. 11 presents the Pareto front solutions obtained from each algorithm. NSGA-II (blue points) demonstrates a well-distributed and optimal set of solutions along the lower-left region of the plot, achieving the best balance between energy efficiency and occupant comfort. MOPSO (orange points) exhibits slightly less diversity but still maintains competitive performance in minimizing energy and PPD. In contrast, QAOA (red points) produces solutions with significantly higher total energy consumption and a wider spread in PPD, indicating a suboptimal trade-off compared to classical methods.

Despite QAOA's limitations in solution quality, it significantly outperformed classical approaches in terms of execution time. QAOA completed the optimization in just 0.54 min, compared to 5.38 min for NSGA-II and 28.14 min for MOPSO for 200 generations. This highlights QAOA's potential for real-time optimization, where rapid decision-making is critical.

The best-performing solutions exhibit low total energy consumption (ranging from 17.84 to 19.84 kWh/m^2/year) and minimal PPD (5.19–5.21 %), reinforcing NSGA-II's ability to effectively balance energy efficiency and occupant comfort. The optimal solutions identified by NSGA-II exhibit several key design characteristics that contribute to improved building performance. Low U-values for the façade (~0.12 W/

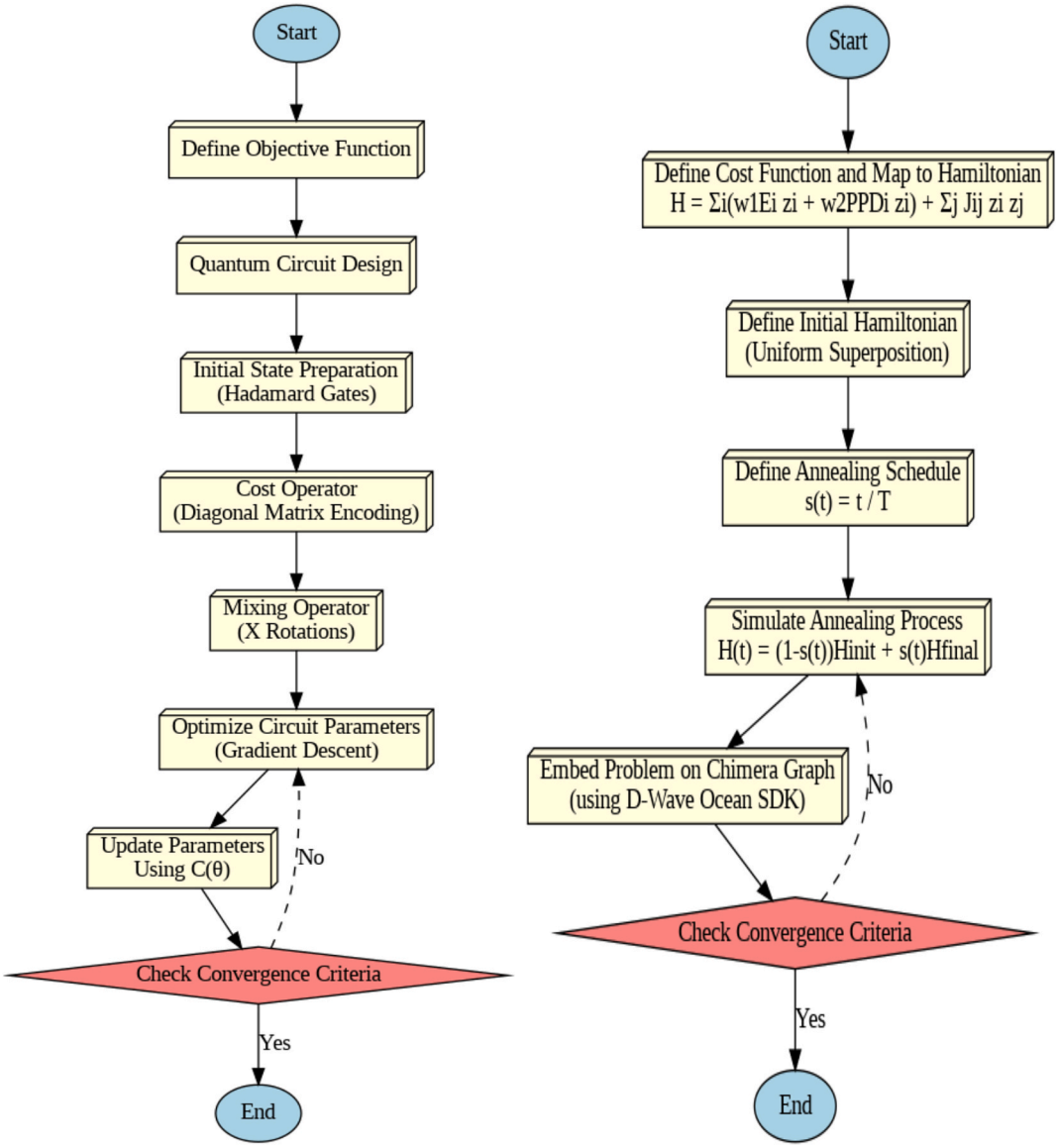


Fig. 7. The left workflow represents the process of solving optimization problems using a Variational Quantum Algorithm (VQA). The right workflow illustrates Quantum Annealing.

Table 5
Machine Learning model performance metrics.

Model	Energy RMSE	Energy MAE	Energy R^2	PPD RMSE	PPD MAE	PPD R^2
Random Forest	1.93	1.26	0.98	0.01	0.01	1.00
Support Vector Regression	11.84	9.86	0.11	6.17	5.23	0.96
K-Nearest Neighbors	12.79	10.54	−0.04	3.32	2.41	0.99
Neural Network	12.30	10.12	0.04	5.10	3.96	0.97
XGBoost	1.89	1.12	0.99	0.013	0.10	1.00

m^2K) ensure effective insulation, minimizing heat losses and enhancing thermal efficiency. A moderate WWR of approximately 34 % strikes a balance between daylight penetration and solar heat gain, preventing excessive overheating while maintaining sufficient natural illumination. Additionally, higher reflectance values (>0.42) contribute to better daylight utilization, reducing the reliance on artificial lighting and improving overall energy efficiency. Internal loads are well-balanced at around 28 W/m^2, optimizing space utilization without generating excessive thermal gains that would increase cooling demand.

To further evaluate the computational efficiency and solution distribution of the optimization algorithms, the number of generations increased from 200 to 500. This adjustment allowed for a more extensive exploration of the solution space, potentially improving the quality of Pareto-optimal solutions. Fig. 12 illustrates the updated Pareto front comparison for NSGA-II, MOPSO, and QAOA after this modification.

With the extended optimization process, NSGA-II maintained its strong performance, continuing to produce well-distributed Pareto-optimal solutions in the low-energy, low-PPD region. MOPSO, however, exhibited a notable change in solution distribution, achieving a broader spread of Pareto-optimal points compared to previous iterations. This suggests that MOPSO benefits from additional generations, enabling it to converge toward improved solutions over time.

In contrast, QAOA maintained its rapid execution speed but did not significantly improve its solution quality. The execution time for NSGA-II increased to 60.18 min, while MOPSO required 147.41 min due to its

Table 6
Example of optimal design configurations identified by NSGA-II for PPD < 10 %.

U-val, facade	WWR	g-value	Internal load	Lighting	Reflectance	Total Energy	Predicted PPD
0.1203	34.347	0.555	28.777	2.000	0.428	19.84	5.195
0.1203	34.348	0.555	28.777	2.000	0.428	19.84	5.195
0.1203	34.440	0.515	28.831	2.056	0.649	19.42	5.197
0.1203	34.440	0.515	28.831	2.056	0.640	19.42	5.197
0.1200	34.430	0.515	28.562	2.017	0.440	19.14	5.200
0.1203	34.440	0.515	28.973	2.055	0.427	19.14	5.200
0.1203	33.427	0.515	28.817	2.055	0.632	18.76	5.202
0.1208	34.438	0.521	27.828	2.011	0.644	18.76	5.202
0.1208	34.366	0.520	29.047	2.011	0.644	18.55	5.204
0.1201	34.418	0.516	28.901	2.000	0.647	17.84	5.212

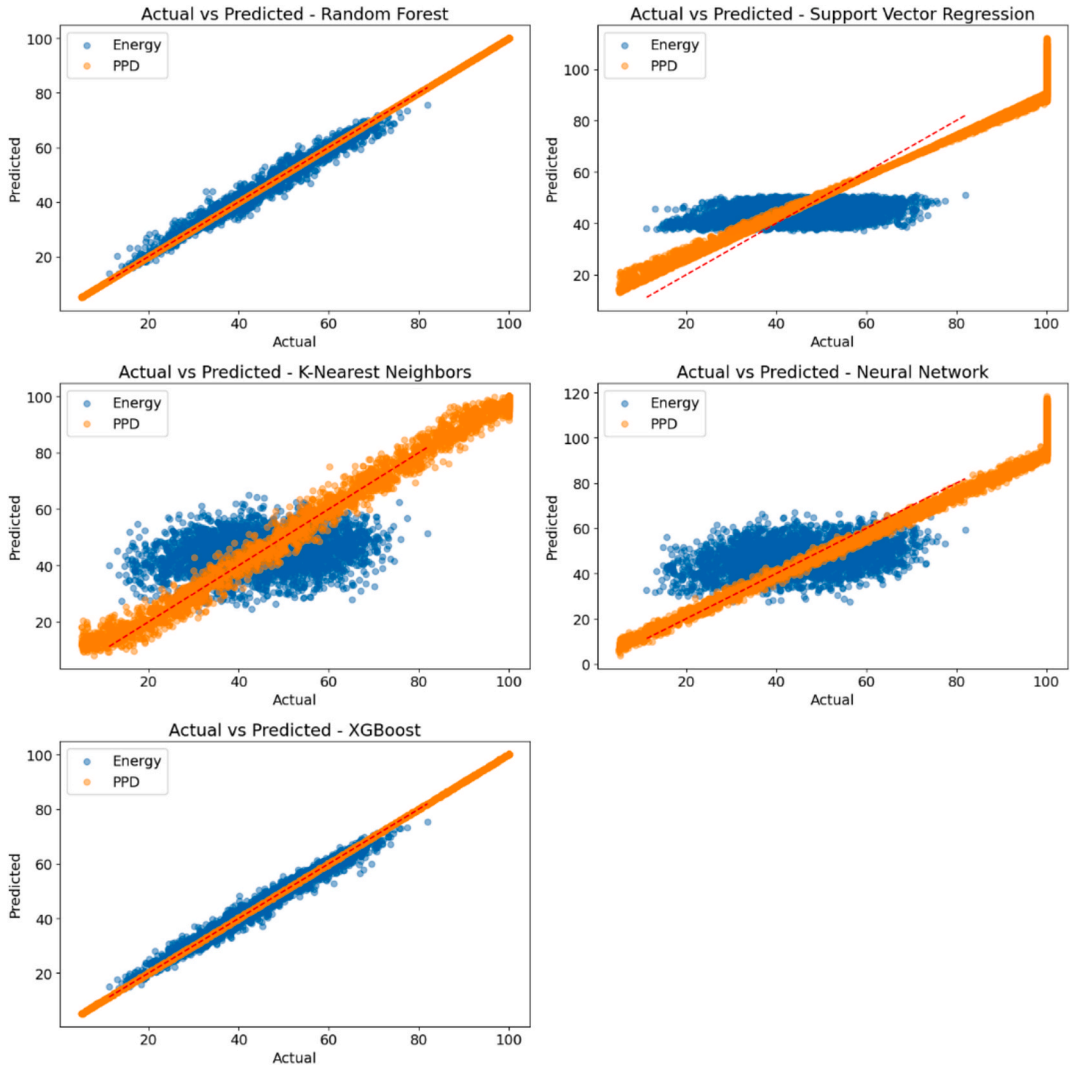


Fig. 8. Scatter plots comparing actual vs. predicted values for total energy consumption and PPD using Random Forest, SVR, KNN, Neural Network, and XGBoost models.

more complex search mechanism. QAOA, however, retained its remarkable computational efficiency, completing the optimization in just 1.73 min, making it nearly 35 times faster than NSGA-II and 85 times faster than MOPSO. The results reaffirm that while quantum-inspired optimization offers substantial speed-ups, classical evolutionary algorithms like NSGA-II and MOPSO remain superior in achieving optimal energy-PPD trade-offs, particularly when given extended computational time.

4.3. Pareto front comparison of NSGA-II, MOPSO, quantum annealing, and quantum processing Unit (QPU)

This study extends the optimization analysis by incorporating Quantum Annealing and Quantum Processing Unit (QPU) solvers alongside NSGA-II and MOPSO, with the objective of evaluating computational efficiency, solution quality, and Pareto front diversity. To ensure a more extensive exploration of the solution space, the number of generations was increased to 500, allowing the algorithms to reach a more stable convergence.

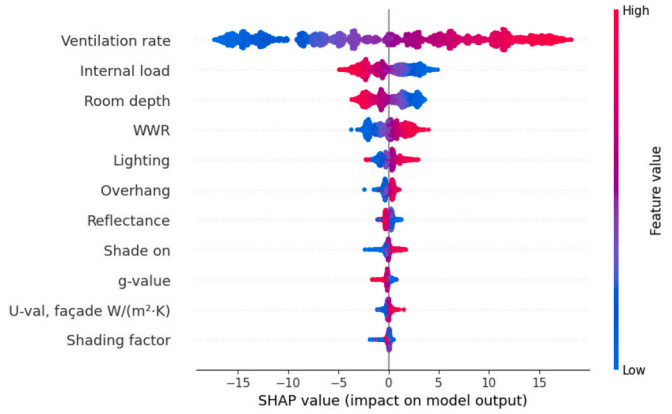


Fig. 9. SHAP summary plot showing the impact of input features on the total energy prediction (kWh/m^2/year) using XGBoost.

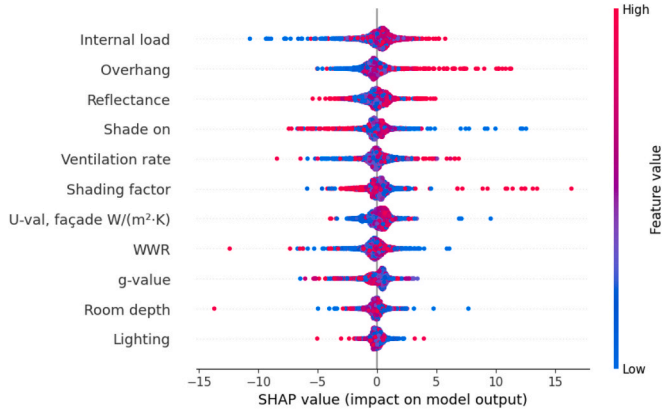


Fig. 10. SHAP summary plot showing the impact of input features on the predicted PPD (%) using XGBoost.

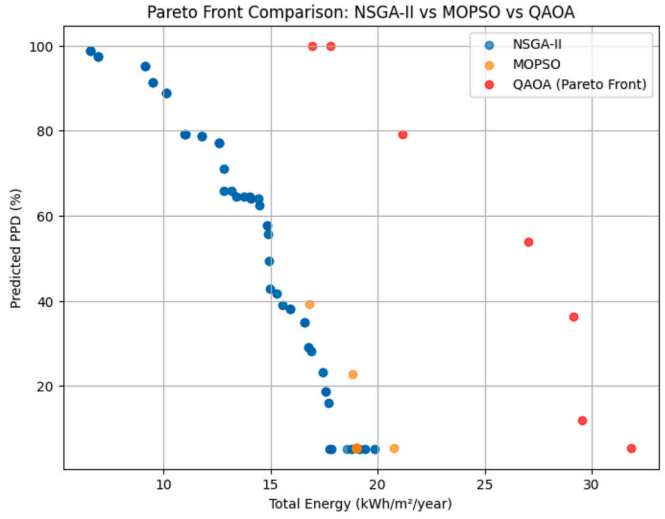


Fig. 11. Pareto front solutions comparing total energy consumption and predicted PPD for NSGA-II (blue), MOPSO (orange), and QAOA (red). The figure highlights the dominance of NSGA-II in generating well-distributed, low-energy and low-PPD solutions. MOPSO presents competitive solutions in the lower range, though less dense.

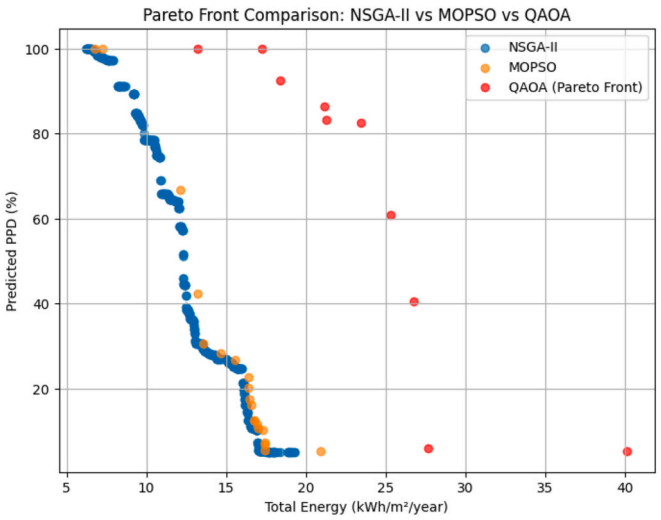


Fig. 12. Pareto front comparison illustrating optimization results using NSGA-II (blue), MOPSO (orange), and QAOA (red) with a generation count of 500. NSGA-II and MOPSO produce densely packed and well-distributed solutions with low energy and low PPD, indicating efficient trade-offs. QAOA solutions appear more scattered and sparse, with relatively higher energy and PPD values, reflecting challenges in convergence and precision within the quantum framework.

The quantum computing aspect of this experiment was designed to run on D-Wave's hybrid solver and QPU (Quantum Processing Unit) using quantum annealing for energy-PPD optimization. The optimization problem was first formulated as a Binary Quadratic Model (BQM), a mathematical structure compatible with quantum annealing solvers. The Leap Hybrid Sampler was employed to execute a hybrid quantum–classical optimization approach, combining quantum sampling with classical pre-processing. Additionally, an attempt was made to execute the optimization directly on D-Wave's Advantage_system7.1 QPU using EmbeddingComposite(DWaveSampler()), which would have enabled quantum annealing on dedicated hardware. However, due to an API authentication issue, the quantum solvers defaulted to a simulator (dimod.ExactSolver), meaning that the results reflect the performance of a quantum annealing simulator rather than real quantum hardware.

The Pareto front solutions obtained from each algorithm are visualized in Fig. 13, illustrating the distribution of solutions generated by NSGA-II (blue), MOPSO (green), Quantum Annealing (red), and QPU-based optimization (purple).

The quantum-inspired methods, Quantum Annealing and QPU-based optimization exhibited a pronounced clustering effect, with solutions concentrated within a narrow range of total energy values. Quantum Annealing demonstrated a bias toward solutions with higher total energy consumption, while QPU-based optimization produced solutions predominantly within the 35–40 kWh/m^2/year range, with PPD values below 10 %. The lack of diversity in these solutions suggests that while quantum solvers can efficiently identify feasible solutions, they currently struggle to maintain a well-spread trade-off distribution, which is essential in multi-objective optimization.

Alongside solution quality, execution time was another critical metric in this comparison. The quantum solvers remained significantly faster, with Quantum Annealing (Hybrid) completing in 0.37 h and the QPU-based optimization finishing in 0.39 h. Hence, the results reinforce the computational speed advantage of quantum computing methods, however, the trade-off between speed and solution quality remains evident, as classical optimization methods still outperform quantum-inspired approaches in maintaining a diverse and well-balanced Pareto front.

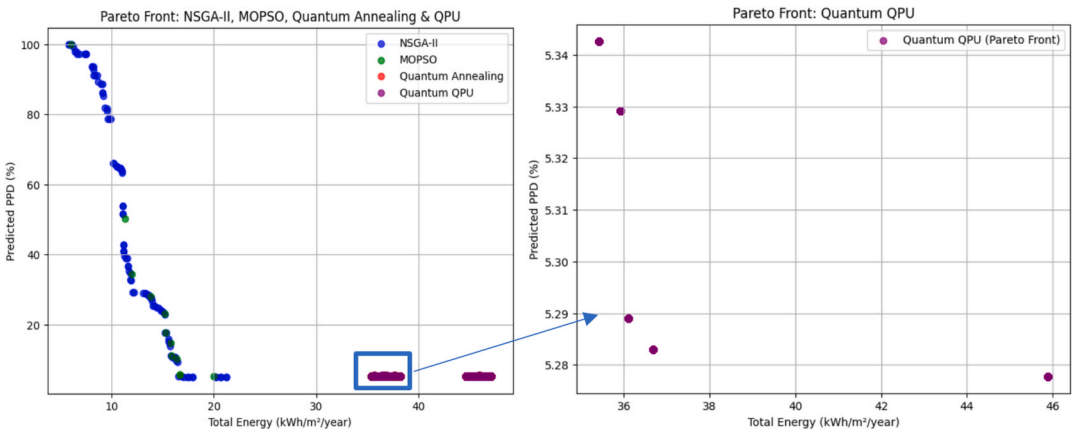


Fig. 13. Pareto front visualization comparing optimization solutions for total energy consumption (kWh/m^2/year) and predicted PPD (%) using NSGA-II (blue), MOPSO (green), Quantum Annealing (red), and Quantum Processing Unit (QPU, purple). The left plot shows the broader trade-off landscape, while the right plot zooms in on QPU performance, revealing a narrow solution range with low PPD but higher energy values compared to classical methods.

4.4. Reflection on the optimization results

The comparison of optimization results across NSGA-II, MOPSO, QAOA, and Quantum Annealing in Fig. 14 and Table 7 highlights distinct trade-offs between solution quality, energy efficiency, and computational speed. The NSGA-II algorithm consistently outperformed other approaches in achieving the lowest total energy consumption, with values ranging from 17.84 to 19.84 kWh/m^2/year, while maintaining PPD values around 5.2 %. These results reinforce the algorithm's ability to navigate the multi-objective optimization space effectively, providing a well-distributed and diverse Pareto front with minimal variation in occupant discomfort.

MOPSO produced solutions that were close to NSGA-II in terms of energy performance, with total energy values around 18.99–20.76 kWh/m^2/year. However, the PPD values were slightly higher, averaging around 5.30 %, indicating that MOPSO's convergence characteristics led to a small trade-off in comfort levels. While still capable of identifying efficient solutions, MOPSO exhibited slightly less precision in balancing the conflicting objectives compared to NSGA-II.

In contrast, QAOA delivered significantly higher energy consumption values, ranging from 31.85 to 55.62 kWh/m^2/year, with a notable increase in PPD, reaching up to 7.81 %. These results indicate that QAOA struggled to maintain a balanced trade-off, favoring solutions that resulted in suboptimal energy performance and occupant comfort. This limitation suggests that while quantum-inspired optimization benefits from computational efficiency, its ability to identify and refine optimal solutions remains inferior to classical evolutionary approaches.

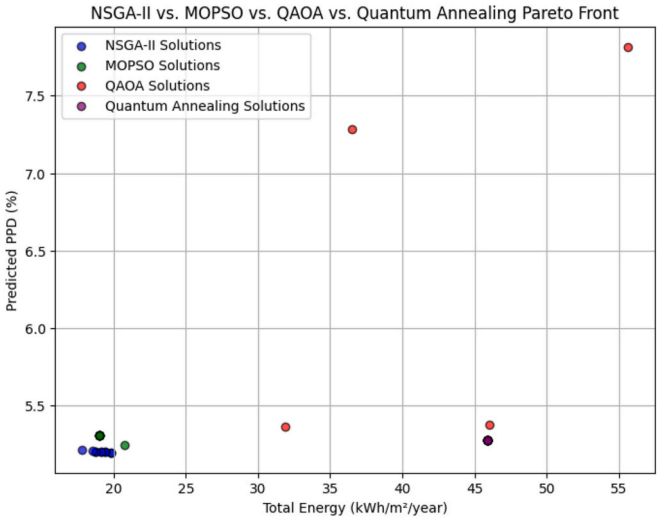


Fig. 14. Pareto front comparison of NSGA-II, MOPSO, QAOA, and Quantum Annealing for solutions with PPD less than 10%. The figure illustrates the trade-offs between total energy consumption and thermal comfort across optimization algorithms. Each point represents a non-dominated solution, with lower-left points indicating better performance in both objectives.

Table 7
Optimization performance ranges for NSGA-II, MOPSO, QAOA, and Quantum Annealing under the constraint of PPD < 10 %. The table presents the spread of total energy consumption and thermal comfort outcomes achieved by each algorithm, highlighting the superior energy efficiency of classical methods compared to quantum-based approaches.

Algorithm	Total Energy (kWh/m^2/year)	Predicted PPD (%)
NSGA-II	17.84 – 19.84	5.19 – 5.21
MOPSO	18.99 – 20.76	5.25 – 5.31
QAOA	31.85 – 55.62	5.36 – 7.81
Quantum Annealing	45.88	5.28

Quantum Annealing exhibited a strong clustering effect, with all solutions concentrated at 45.88 kWh/m^2/year and a PPD of 5.27 %. This lack of diversity in the Pareto front underscores a critical limitation in the current implementation of quantum solvers, as they fail to explore the full range of optimal trade-offs. While Quantum Annealing offers rapid computation, the results demonstrate that it lacks the flexibility required to generate a comprehensive set of energy-efficient configurations.

4.5. Optimization results and statistical comparison

To ensure fair and consistent benchmarking of all algorithms, the optimization experiments were conducted under identical evaluation budgets. Specifically, each classical algorithm, NSGA-II and MOPSO, was run with a population size of 50, across 100 generations, and repeated for 20 independent trials, yielding 5000 fitness evaluations per trial and a total of 100,000 evaluations per algorithm. The same number of evaluations was mirrored in the quantum-inspired methods (QAOA and QA) through equivalent iteration limits to maintain computational parity. This approach aligns with best practices in multi-objective optimization research, where the number of evaluations directly influences algorithmic performance and comparability.

To quantify and compare optimization outcomes, two performance indicators were employed, Hypervolume (HV) [55,56] and Inverted Generational Distance (IGD) [57,58]. Hypervolume reflects both

convergence to the Pareto-optimal front and diversity of solutions by measuring the volume of the objective space dominated by the solution set. In contrast, IGD evaluates the closeness of the obtained solutions to a known reference front, with lower values indicating higher convergence accuracy. These two indicators, taken together, offer a comprehensive assessment of an algorithm's ability to find diverse, high-quality solutions.

Table 8 presents the hypervolume results for the four tested algorithms. NSGA-II attained the highest mean HV value (6696.22) with very low standard deviation (±34.36), reflecting stable performance across multiple runs and a consistent ability to find diverse and expansive Pareto fronts. MOPSO followed with a lower mean HV (6254.73) and greater variability (±202.64), implying that while it can find good solutions, its performance is more sensitive to initial conditions or parameter settings. QAOA achieved a moderate mean HV (5195.55 ± 147.24), demonstrating some ability to explore the search space but not as effectively as the classical algorithms. Quantum Annealing (QA), in contrast, showed a very low HV (1422.91 ± 0.00), meaning its solutions collapsed into a small region with no diversity, indicating poor exploration.

To further assess convergence accuracy, Table 9 presents the IGD values for the same runs. IGD is a convergence metric where lower values indicate that the solutions are both accurate and well-aligned with the true optimal front. It is calculated by averaging the distance from each point on the reference front to its nearest neighbor in the obtained solution set. This means that a low IGD implies that the algorithm found diverse solutions but also captured the shape of the true Pareto front effectively.

NSGA-II demonstrated the best performance, achieving the lowest mean IGD of 1.2 with a very small standard deviation of 0.3. This confirms that NSGA-II consistently generated solutions that are close to the ideal front with minimal variability. This result is also evident in Fig. 14, where NSGA-II points are densely packed in the lower-left corner of the plot, representing the optimal trade-off between energy use and predicted PPD. Moreover, MOPSO showed a higher mean IGD of 3.5 and a standard deviation of 0.8. Although not as precise as NSGA-II, MOPSO still managed to populate the optimal region of the Pareto front with reasonable convergence.

QAOA exhibited the highest mean IGD of 14.7 and a large standard deviation of 7.5. As seen in Fig. 14, the QAOA solutions are spread out and include points that are far from the optimal region. Some solutions are in the top-right area of the plot with high energy and high predicted PPD, which significantly increases the IGD. Although one QAOA point lies closer to the optimal region, the presence of distant, suboptimal solutions heavily penalizes the average, highlighting the algorithm's lack of consistent convergence behavior. Furthermore, Quantum Annealing had a mean IGD of 10.8 with a standard deviation of 5.9. Its performance was slightly better than QAOA but still far from ideal. In Fig. 14, the QA solutions cluster together but are located in a remote part of the objective space, around 45 kW-hours per square meter per year, indicating poor alignment with the reference front. The compactness of this cluster reduces variability, but the large distance from the optimal region results in a high IGD.

To assess whether the observed differences in performance between the algorithms were meaningful or simply due to random variation, the Friedman test [59,88] was applied (Table 10). This test is a non-parametric statistical method used to compare more than two algorithms across multiple datasets or runs. It ranks the algorithms for each run and then checks whether the differences in their rankings are statistically significant. In this case, the Chi-squared values [89] χ^2 (28.92 for HV and 17.40 for IGD) indicate how much the rankings vary across the algorithms. The associated p-values are both extremely small, far below the common significance threshold of 0.05. This means there is strong evidence that at least one algorithm performs differently from the others in a statistically significant way.

To understand how each optimization algorithm compares with the others, we used the Nemenyi post-hoc test [60,90] following the Friedman test. While the Friedman test confirmed that the differences in performance across the algorithms are statistically significant, it does not indicate where these differences occur. The Nemenyi test fills that gap by performing pairwise comparisons to reveal which specific algorithms differ significantly from each other in terms of performance.

Starting with the HV results (Table 11), the Nemenyi test shows that NSGA-II significantly outperforms all other algorithms. This includes highly significant differences between NSGA-II and both QAOA and Quantum Annealing, with p-values of 0.0012 and 0.000001 respectively, indicating a very low probability that these differences are due to chance. Even the difference between NSGA-II and MOPSO is statistically significant ($p = 0.043$), although the margin is smaller, reflecting that MOPSO is competitive but still inferior. Furthermore, MOPSO is significantly better than Quantum Annealing ($p = 0.00039$) and better than QAOA ($p = 0.029$), suggesting that MOPSO's solutions are more diverse and better spread along the Pareto front. The comparison between QAOA and Quantum Annealing also revealed a statistically significant difference ($p = 0.0051$), indicating that QAOA, despite its flaws, still offers broader solution coverage than Quantum Annealing, which tends to collapse into a small region of the solution space.

In terms of IGD results (Table 12), which measure how close the obtained solutions are to the true optimal Pareto front, NSGA-II again stands out, performing significantly better than all other algorithms. The differences between NSGA-II and QAOA ($p = 0.000002$), and NSGA-II and QA ($p = 0.003$) are both statistically strong. MOPSO also performs better than QAOA ($p = 0.00035$) and has a slight, though not statistically strong, edge over Quantum Annealing ($p = 0.078$). The difference between QAOA and QA is significant ($p = 0.012$), indicating that QAOA achieves better convergence even if its spread (HV) is limited. This result aligns with the visual analysis of the solution sets, QAOA tends to produce solutions that are closer to the optimal front but

Table 8
Hypervolume (HV) Results.

Algorithm	Mean HV	Std. Dev
NSGA-II	6696.22	34.36
MOPSO	6254.73	202.64
QAOA	5195.55	147.24
QA	1422.91	0.00

Table 9
Inverted Generational Distance (IGD) Results.

Algorithm	Mean IGD	Std. Dev
NSGA-II	1.2	0.3
MOPSO	3.5	0.8
QAOA	14.7	7.5
Quantum Annealing	10.8	5.9

Table 10
Friedman Test Results.

Metric	χ^2	p-value
HV	28.92	0.00000233
IGD	17.40	0.00058

Table 11
Nemenyi Post-hoc Test for HV.

	NSGA-II	MOPSO	QAOA	QA
NSGA-II	1.000	0.043	0.0012	0.000001
MOPSO	0.043	1.000	0.029	0.00039
QAOA	0.0012	0.029	1.000	0.0051
QA	0.000001	0.00039	0.0051	1.000

Table 12
Nemenyi Post-hoc Test for IGD.

	NSGA-II	MOPSO	QAOA	QA
NSGA-II	1.000	0.041	0.000002	0.003
MOPSO	0.041	1.000	0.00035	0.078
QAOA	0.000002	0.00035	1.000	0.012
QA	0.003	0.078	0.012	1.000

are scattered inconsistently, while QA solutions are both limited in spread and farther from the front.

4.6. Convergence and variance in optimization results

Beyond the Pareto front analysis, an investigation into the convergence behavior and variance of final solutions provides additional insights into the stability and robustness of the optimization approaches. The convergence trends of total energy and PPD for NSGA-II, MOPSO, and QAOA are depicted in Fig. 15, while the variance in final Pareto front solutions is presented in Fig. 16. These analyses highlight how each algorithm progresses toward optimal solutions over iterations and the degree of consistency in their final results.

As seen in Fig. 15, NSGA-II and MOPSO exhibit stable and well-defined convergence patterns for both energy and PPD. NSGA-II demonstrates a smooth reduction in total energy values over iterations, consistently reaching solutions below 20 kWh/m^2/year. MOPSO follows a similar trend but converges at slightly higher energy values. The PPD convergence patterns reinforce these findings, with MOPSO reaching a stable region within the first 20 iterations, while NSGA-II continues refining solutions over more iterations. This suggests that NSGA-II has a stronger ability to explore the solution space, leading to improved energy-PPD trade-offs. QAOA, however, shows significantly different behavior, with energy values fluctuating around 100 kWh/m^2/year, indicating a lack of effective convergence.

The variance analysis in Fig. 16 further confirms these observations. NSGA-II exhibits the lowest variance in total energy values, with a variance of 31.47, demonstrating its ability to consistently find optimal energy-efficient solutions. However, its PPD variance is the highest (1302.55), reflecting the algorithm's extensive search for diverse comfort levels. MOPSO shows moderate variance in both energy (12.39) and PPD (732.39), indicating a stable but slightly less refined exploration of the Pareto front. QAOA, in contrast, exhibits the highest variance in total energy (121.42) and significant fluctuation in PPD values (1201.90), highlighting its instability in balancing energy efficiency and comfort.

These results reinforce the superior stability and efficiency of NSGA-II in optimizing building performance, followed by MOPSO as a competitive alternative. QAOA, despite its computational speed advantage, struggles to maintain solution consistency, making it less reliable for real-world applications where predictability is crucial. The findings emphasize the importance of combining fast quantum-inspired optimization with more robust classical approaches to achieve optimal trade-offs in building energy and occupant comfort.

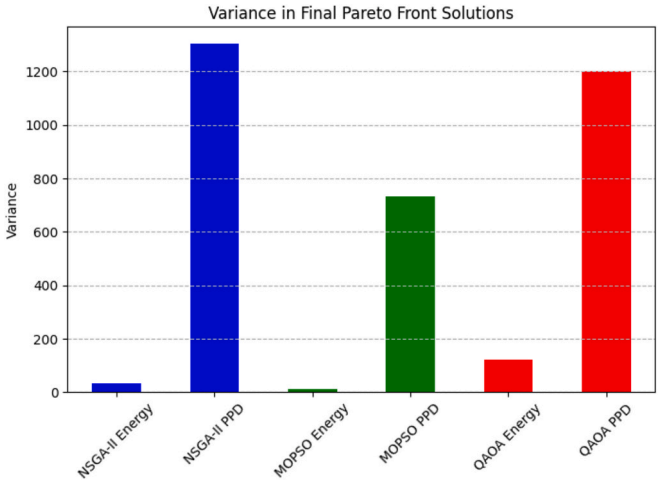


Fig. 16. Variance of final Pareto front solutions for NSGA-II, MOPSO, and QAOA across total energy and PPD objectives. NSGA-II presents the smallest energy variance, indicating high stability in energy optimization, while its PPD variance is the largest, suggesting active exploration across thermal comfort levels. MOPSO shows moderate variance in both metrics, whereas QAOA demonstrates significant variability in PPD with limited diversity in energy outcomes.

4.7. Real-time convergence of NSGA-II and QAOA

To better understand the dynamic optimization process of NSGA-II and QAOA, a real-time convergence analysis was conducted, where the progression of energy and PPD values over iterations was tracked.

The energy convergence plot in Fig. 17 (left) highlights distinct behaviors in how NSGA-II and QAOA refine their solutions. NSGA-II follows a gradual but stable improvement trajectory, progressively reducing energy consumption over iterations, whereas QAOA exhibits highly unstable fluctuations in the early stages before settling into a more stable range. This suggests that QAOA's initial search space exploration is erratic, likely due to its probabilistic nature, which can lead to randomly sampled suboptimal solutions before convergence

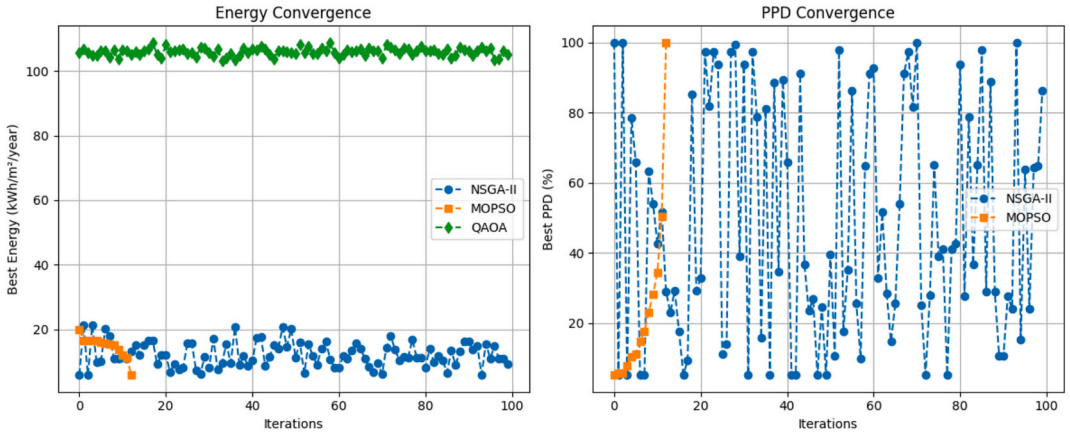


Fig. 15. Convergence behavior of the optimization algorithms across 100 iterations. The left plot illustrates energy convergence, where NSGA-II consistently reaches the lowest energy values, followed by MOPSO with moderate improvement, and QAOA stagnating at higher values. The right plot presents convergence of predicted PPD, where NSGA-II demonstrates iterative refinement, MOPSO stabilizes quickly, and QAOA data is unavailable due to unstable results.

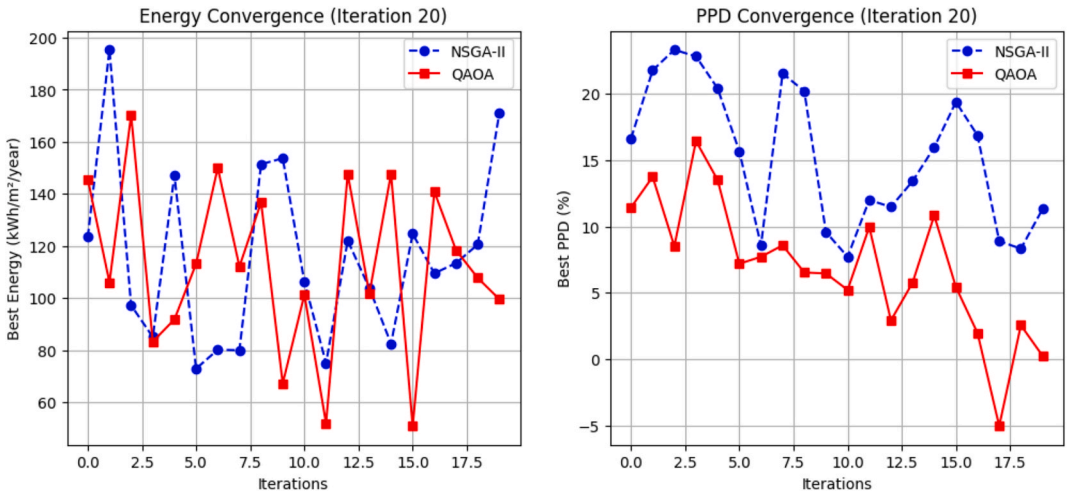


Fig. 17. Real-Time Convergence of NSGA-II and QAOA. Left: Energy convergence over iterations, showing NSGA-II's stable improvement versus QAOA's early fluctuations. Right: PPD convergence, highlighting NSGA-II's gradual optimization compared to QAOA's erratic early-stage behavior before stabilizing.

occurs.

Similarly, the PPD convergence plot in Fig. 17 (right) reveals NSGA-II's characteristic of steady optimization, refining comfort-related metrics over iterations. QAOA, in contrast, shows an irregular pattern in its early iterations, reflecting its higher uncertainty in identifying optimal comfort conditions. However, as iterations progress, QAOA stabilizes at a lower PPD range, potentially indicating that its faster computational approach allows it to reach an acceptable solution sooner, albeit with less precision.

This real-time analysis suggests that while QAOA is capable of quickly narrowing down solutions, its early instability introduces a risk of suboptimal convergence, whereas NSGA-II demonstrates superior robustness and consistency throughout the optimization process. The ability to track these iterations in real time provides valuable insight into how different algorithms behave dynamically, reinforcing the importance of balancing computational speed with solution reliability.

4.8. Hypervolume-based convergence analysis

Evaluating the performance of multi-objective optimization algorithms requires more than examining individual fitness values such as energy consumption and predicted percentage of dissatisfied occupants. These objective-specific indicators provide valuable insight, but they do not reflect how well an algorithm handles the trade-off between competing goals. Hypervolume offers a more comprehensive performance metric because it combines information about both convergence to the optimal front and the diversity of the solution set.

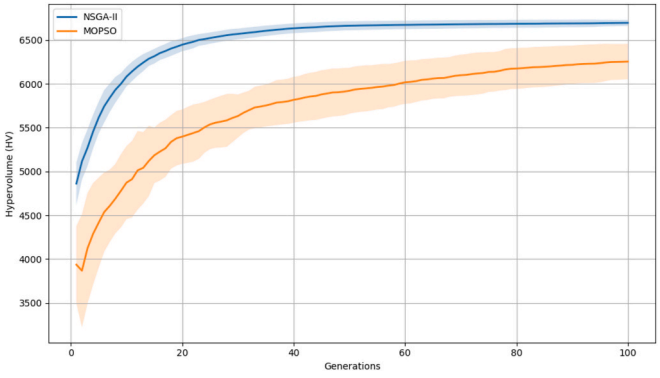


Fig. 18. Hypervolume convergence curves over generations for NSGA II and MOPSO. NSGA II shows faster and more consistent improvement, achieving better convergence and wider solution diversity compared to MOPSO.

Fig. 18 shows how the hypervolume values evolve over generations for NSGA II and MOPSO, using an equal number of total fitness evaluations. The curve for NSGA II demonstrates a steady and strong improvement, reaching higher hypervolume values across all generations. This reflects its ability to generate a wide and well-distributed set of high-quality solutions throughout the optimization process. MOPSO also improves over time but at a slower rate, achieving lower final hypervolume values and showing more variability.

Quantum Approximate Optimization Algorithm and Quantum Annealing were excluded from this comparison because their evaluation structure differs significantly from NSGA II and MOPSO. They rely on discrete quantum-inspired sampling strategies that do not allow a comparable generation-based progression or consistent evaluation budget. Including them in the hypervolume convergence comparison would not provide a fair or meaningful interpretation.

4.9. Surrogate model validation after optimization

After completing the optimization process using the XGBoost surrogate model, it is essential to validate the accuracy of the surrogate's predictions. Although the surrogate was trained on a large dataset of 15,000 IDA ICE simulations, the optimization algorithms relied only on the learned model, not the physical simulation engine. To ensure that the surrogate-based optimization results are physically meaningful and accurate, a post-evaluation step was conducted using the original simulation tool.

In this step, a set of non-dominated solutions from each optimization algorithm was selected and re-evaluated using IDA ICE. This allowed a direct comparison between the surrogate predictions and the actual simulation outcomes for both total energy and PPD. The goal is to verify whether the surrogate accurately captured the underlying simulation behavior, especially in the optimal regions suggested by the optimization.

As shown in Table 13, the results confirm a strong agreement between the surrogate and simulation values for NSGA-II and MOPSO,

Table 13
Surrogate vs Simulation Validation for Non-Dominated Solutions.

Algorithm	MAE Energy (kWh/m^2/year)	MAE PPD (%)
NSGA-II	0.72	1.34
MOPSO	1.05	2.11
QAOA	—	—
QA	—	—

with very small mean absolute errors (MAE) for both total energy and PPD. This demonstrates the reliability of the XGBoost surrogate model for predicting high-quality solutions in the explored design space. QAOA and QA were excluded from this validation due to their instability and poor convergence observed in earlier sections.

4.10. Reflections and future directions

This study underscores both the promise and current limitations of quantum computing in building performance optimization. Classical evolutionary algorithms such as NSGA-II and MOPSO consistently outperformed their quantum-inspired counterparts in terms of convergence stability, diversity, and overall solution quality. These results are consistent with findings from benchmarking studies like [91], which report the instability and underperformance of QAOA and quantum annealing under current technological constraints. Nonetheless, quantum computing remains a promising paradigm, offering theoretical advantages in exploring high-dimensional solution spaces more efficiently than classical methods.

A key obstacle encountered in this study was the inability to access functional quantum hardware. All quantum annealing experiments were simulated due to API limitations, precluding execution on real D-Wave processors. Even with access, today's quantum machines are constrained by a limited number of qubits, poor connectivity, and high susceptibility to noise and decoherence—challenges well-documented in [92]. These hardware constraints reduce the reliability and repeatability of quantum optimization in complex, multi-objective applications such as balancing energy use and occupant comfort. For quantum computing to gain practical relevance in this domain, significant advancements in qubit stability, noise reduction, and error correction are required. Future research should therefore aim to test on physical quantum hardware when accessible, apply automated tuning strategies to improve QAOA parameterization, and enhance QUBO transformation techniques to preserve the structure of real-world optimization problems.

Beyond hardware limitations, the structure and configuration of current quantum algorithms contribute to their performance gaps. QAOA's effectiveness is highly sensitive to the choice of initial parameters and circuit depth, which remains fixed and does not scale with problem complexity. This rigidity restricts the algorithm's flexibility and may cause premature convergence to suboptimal solutions. Introducing adaptive-depth schemes that adjust based on intermediate feedback could improve solution quality. Similarly, quantum annealing's dependence on the QUBO formulation can oversimplify multi-objective problems, reducing the expressiveness of the optimization landscape. These constraints limit the algorithms' ability to produce well-distributed Pareto fronts in high-dimensional spaces like those found in building performance modeling.

To overcome these challenges, refinement at the algorithmic level is essential. QAOA could benefit from *meta*-learning or reinforcement learning approaches to enhance parameter tuning dynamically. Post-processing methods using classical models, such as surrogate filtering or deep learning refinement, can further improve quantum outputs by mitigating clustering effects, particularly in quantum annealing where solution diversity was often compromised. Such enhancements will be key to improving convergence reliability and broadening the application range of quantum optimization.

As the field progresses, quantum computing is expected to play an increasingly important role in real-time building control and large-scale energy optimization. For instance, quantum techniques may enable responsive adjustment of HVAC, lighting, and shading systems based on environmental inputs, offering significant advantages in smart building contexts. This potential was explored by [93], who demonstrated quantum-based improvements in building-to-grid energy coordination, reporting speed gains over classical solvers. Applications could also extend to multi-scale urban energy networks and advanced digital twin environments, where continuous, data-driven optimization across systems and time scales is required.

Given the current strengths of classical algorithms and the emerging capabilities of quantum solvers, a hybrid quantum–classical approach appears most viable. Quantum algorithms can serve as pre-screening mechanisms, quickly identifying promising solution regions in large, complex spaces. These regions can then be fine-tuned using stable classical methods like NSGA-II or MOPSO. This layered strategy aligns with the findings of [94], who presented a software tool integrating quantum and classical methods for improved optimization outcomes. Reinforcement learning may also guide when and how to apply each method during different optimization phases, maximizing efficiency and effectiveness.

Beyond theoretical exploration, practical integration of these optimization frameworks into real-world systems is critical. Surrogate models such as XGBoost, once trained on simulation data, can be embedded into digital twins or building management systems (BMS) for rapid performance predictions. Optimization algorithms like NSGA-II or hybrid quantum–classical strategies can then operate in near real-time to adjust operational parameters such as HVAC control or shading systems based on sensor inputs. This enables data-driven, adaptive control of energy consumption and comfort. With modern BMS platforms increasingly supporting modular APIs and real-time data streams, such integrations are technically feasible and can significantly improve the responsiveness and efficiency of smart buildings.

While quantum computing is not yet positioned to replace classical approaches in building performance optimization, it holds significant long-term potential. As emphasized by [95], future progress depends on advances in hardware scalability, robust error correction, and algorithmic innovation. This study demonstrates that classical methods remain dominant for practical deployment today, but as quantum technologies evolve, their integration into hybrid frameworks could redefine the optimization landscape for energy-efficient and occupant-centric building design.

5. Conclusion

This study explored the use of classical and quantum optimization techniques to enhance building performance, with a focus on reducing energy consumption and improving occupant comfort. The comparative analysis of NSGA-II, MOPSO, QAOA, and quantum annealing highlighted distinct strengths and limitations. Classical algorithms, particularly NSGA-II and MOPSO, exhibited robust convergence, stability, and diverse solution sets, confirming their suitability for practical multi-objective optimization in building applications. Quantum methods demonstrated considerably faster execution times, yet their performance was limited by unstable convergence and narrow solution spaces. These limitations reflect current algorithmic constraints rather than fundamental shortcomings of the approach. Nonetheless, the observed speed advantages indicate potential value for rapid solution generation in time-sensitive scenarios. A promising direction is the integration of quantum techniques within hybrid frameworks that combine quantum preprocessing with classical optimization. Such strategies could balance computational speed and solution quality, offering new possibilities for real-time building performance management. Future research should focus on advancing these hybrid workflows, improving algorithmic robustness, and evaluating scalability in operational environments. While classical approaches currently offer more reliable outcomes, quantum-assisted optimization may become increasingly relevant as methods mature, contributing to more adaptive and efficient building management systems.

Submission for Special Issue:

"Decarbonising the Built Environment: Innovations and Strategies for a Sustainable Future"

Funding

The authors gratefully acknowledge OsloMet – Oslo Metropolitan University for covering the article processing charges (APC) for this publication.

CRediT authorship contribution statement

Haidar Hosamo: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Data curation, Conceptualization. **Vagelis Plevris:** Writing – review & editing, Writing – original draft, Validation, Methodology, Investigation, Data curation, Conceptualization. **Dimitrios Kraniotis:** Writing – review & editing, Writing – original draft, Methodology, Investigation, Data curation, Conceptualization. **Christian Nordahl Rolfsen:** Writing – review & editing, Writing – original draft, Methodology, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] F.S. Hafez, et al., Energy efficiency in sustainable buildings: a systematic review with taxonomy, challenges, motivations, methodological aspects, recommendations, and pathways for future research, Energy Strategy Rev. 45 (Jan. 2023) 101013, https://doi.org/10.1016/j.esr.2022.101013.

[2] H.H. Hosamo, M.S. Tingstveit, H.K. Nielsen, P.R. Svennevig, K. Svidt, Multiobjective optimization of building energy consumption and thermal comfort based on integrated BIM framework with machine learning-NSGA II, Energy Build. 277 (Dec. 2022) 112479, https://doi.org/10.1016/j.enbuild.2022.112479.

[3] H.H. Hosamo, H.K. Nielsen, D. Kraniotis, P.R. Svennevig, K. Svidt, Improving building occupant comfort through a digital twin approach: a Bayesian network model and predictive maintenance method, Energy Build. 288 (Jun. 2023) 112992, https://doi.org/10.1016/j.enbuild.2023.112992.

[4] S. Salimi, E. Estrella Guillén, H. Samuelson, Exceedance Degree-Hours: a new method for assessing long-term thermal conditions, Indoor Air 31 (6) (2021) 2296–2311, https://doi.org/10.1111/ina.12855.

[5] H. Hosamo, S. Mazzetto, Performance evaluation of machine learning models for predicting energy consumption and occupant dissatisfaction in buildings, Buildings 15 (1) (2025) 1, https://doi.org/10.3390/buildings15010039.

[6] A.G. Gad, Particle swarm optimization algorithm and its applications: a systematic review, Arch. Comput. Methods Eng. 29 (5) (Aug. 2022) 2531–2561, https://doi.org/10.1007/s11831-021-09694-4.

[7] V. Plevris, M. Papadrakakis, A hybrid particle swarm—gradient algorithm for global structural optimization, Comput.-Aided Civ. Infrastruct. Eng. 26 (1) (2011) 48–68, https://doi.org/10.1111/j.1467-8667.2010.00664.x.

[8] N.D. Lagaros, V. Plevris, N. Ath Kallioras, The Mosaic of metaheuristic algorithms in structural optimization, Arch. Comput. Methods Eng. 29 (7) (Nov. 2022) 5457–5492, https://doi.org/10.1007/s11831-022-09773-0.

[9] K. Cao, J. Wu, Q. Huang, and Y. Gan, 'Optimization Study of KNN Classification Algorithm on Large-Scale Datasets: Real-Time Optimization Strategy Based on Balanced KD Tree and Multi-threaded Parallel Computing', in *2023 4th International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI)*, Aug. 2023, pp. 423–427. https://doi.org/10.1109/ICHCI58871.2023.10277807.

[10] G. Feng, Feature selection algorithm based on optimized genetic algorithm and the application in high-dimensional data processing, PLoS One 19 (5) (May 2024) e0303088, https://doi.org/10.1371/journal.pone.0303088.

[11] L. Liao, Distributed data real-time transaction calculation based on collaborative optimization and multi-objective genetic algorithm, Int. J. Intell. Inf. Technol. 20 (1) (2023) 1–16, https://doi.org/10.4018/IJIIT.333632.

[12] L. Zhang, et al., A review of machine learning in building load prediction, Appl. Energy 285 (Mar. 2021) 116452, https://doi.org/10.1016/j.apenergy.2021.116452.

[13] Y. Zhao, S. Qian, C. Jing, Prediction and optimization of indoor thermal environment and energy consumption based on artificial neural network, IOP Conf. Ser.: Earth Environ. Sci 463 (1) (Mar. 2020) 012028, https://doi.org/10.1088/1755-1315/463/1/012028.

[14] T.V. Quang, D.T. Doan, N.L. Phuong, G.Y. Yun, Data-driven prediction of indoor airflow distribution in naturally ventilated residential buildings using combined CFD simulation and machine learning (ML) approach, J. Build. Phys. 47 (4) (Jan. 2024) 439–471, https://doi.org/10.1177/17442591231219025.

[15] A. Ahmadi, Quantum computing and artificial intelligence: the synergy of two revolutionary technologies, Asian J. Electr. Sci. 12 (2) (2023) 2, https://doi.org/10.51983/ajes-2023.12.2.4118.

[16] X. Zhao, et al., Unraveling quantum computing system architectures: an extensive survey of cutting-edge paradigms, Inf. Softw. Technol. 167 (Mar. 2024) 107380, https://doi.org/10.1016/j.infsof.2023.107380.

[17] T.Q. Duong, L.D. Nguyen, B. Narottama, J.A. Ansere, D.V. Huynh, H. Shin, Quantum-inspired real-time optimization for 6G networks: opportunities, challenges, and the road ahead, IEEE Open J. Commun. Soc. 3 (2022) 1347–1359, https://doi.org/10.1109/OJCOMS.2022.3195219.

[18] Y.-J. Chang, C.-F. Nien, K.-P. Huang, Y.-T. Zhang, C.-H. Cho, C.-R. Chang, Quantum Computing for Optimization with Ising Machine, IEEE Nanotechnol. Mag. 18 (3) (Jun. 2024) 15–22, https://doi.org/10.1109/MNANO.2024.3378485.

[19] J.M. Arrazola, A. Delgado, B.R. Bardhan, S. Lloyd, Quantum-inspired algorithms in practice, Quantum 4 (Aug. 2020) 307, https://doi.org/10.22331/q-2020-08-13-307.

[20] B. Song, 'Feasibility of Quantum Genetic Algorithm in Optimizing Construction Scheduling', Delft University of Technology, Delft, Netherlands, 2013. [Online]. Available: https://resolver.tudelft.nl/uuid:a3c8bdc4-af73-442e-9331-f536c32dc971.

[21] Y. Wang, J.E. Kim, K. Suresh, Opportunities and challenges of quantum computing for engineering optimization, J. Comput. Inf. Sci. Eng. 23 (060817) (2023) Aug, https://doi.org/10.1115/1.4062969.

[22] X. Zhou, H. Du, S. Xue, Z. Ma, Recent advances in data mining and machine learning for enhanced building energy management, Energy 307 (Oct. 2024) 132636, https://doi.org/10.1016/j.energy.2024.132636.

[23] Y. Chen, M. Guo, Z. Chen, Z. Chen, Y. Ji, Physical energy and data-driven models in building energy prediction: a review, Energy Rep. 8 (Nov. 2022) 2656–2671, https://doi.org/10.1016/j.egyr.2022.01.162.

[24] D. Ramos, P. Faria, A. Morais, Z. Vale, Using decision tree to select forecasting algorithms in distinct electricity consumption context of an office building, Energy Rep. 8 (Jun. 2022) 417–422, https://doi.org/10.1016/j.egyr.2022.01.046.

[25] J. Zhou, S. Zhu, Y. Qiu, D.J. Armaghani, A. Zhou, W. Yong, Predicting tunnel squeezing using support vector machine optimized by whale optimization algorithm, Acta Geotech. 17 (4) (Apr. 2022) 1343–1366, https://doi.org/10.1007/s11440-022-01450-7.

[26] A. Bampoulas, F. Pallonetto, E. Mangina, D.P. Finn, An ensemble learning-based framework for assessing the energy flexibility of residential buildings with multicomponent energy systems, Appl. Energy 315 (Jun. 2022) 118947, https://doi.org/10.1016/j.apenergy.2022.118947.

[27] G. Gokhale, B. Claessens, C. Develder, Physics informed neural networks for control oriented thermal modeling of buildings, Appl. Energy 314 (May 2022) 118852, https://doi.org/10.1016/j.apenergy.2022.118852.

[28] J. Kallio, J. Tervonen, P. Räsänen, R. Mäkynen, J. Koivusaari, J. Peltola, Forecasting office indoor CO2 concentration using machine learning with a one-year dataset, Build. Environ. 187 (Jan. 2021) 107409, https://doi.org/10.1016/j.buildenv.2020.107409.

[29] X. Liang, J. Shim, O. Anderton, D. Song, Low-cost data-driven estimation of indoor occupancy based on carbon dioxide (CO2) concentration: a multi-scenario case study, J. Build. Eng. 82 (Apr. 2024) 108180, https://doi.org/10.1016/j.jobe.2023.108180.

[30] Z. Zhang, X. Zhu, and D. Liu, 'Model of Gradient Boosting Random Forest Prediction', in *2022 IEEE International Conference on Networking, Sensing and Control (ICNSC)*, Dec. 2022, pp. 1–6 https://doi.org/10.1109/ICNSC55942.2022.10004112.

[31] S. Ferreira, B. Gunay, A. Wills, F. Rizvi, A neural network-based surrogate model to predict building features from heating and cooling load signatures, J. Build. Perform. Simul. 17 (5) (Sep. 2024) 631–654, https://doi.org/10.1080/19401493.2024.2375304.

[32] M.K. Ngueajio, G. Washington, D.B. Rawat, Y. Ngueabou, Intrusion Detection Systems using support Vector Machines on the KDDCUP'99 and NSL-KDD Datasets: a Comprehensive Survey, in: K. Arai (Ed.), *Intelligent Systems and Applications*, Springer International Publishing, Cham, 2023, pp. 609–629, https://doi.org/10.1007/978-3-031-16078-3_42.

[33] K. Parvin, et al., Intelligent Controllers and Optimization Algorithms for Building Energy Management Towards Achieving Sustainable Development: challenges and prospects, IEEE Access 9 (2021) 41577–41602, https://doi.org/10.1109/ACCESS.2021.3065087.

[34] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (Apr. 2002) 182–197, https://doi.org/10.1109/4235.996017.

[35] B. Chen, et al., Multiobjective optimization of building energy consumption based on BIM-DB and LSSVM-NSGA-II, J. Clean. Prod. 294 (Apr. 2021) 126153, https://doi.org/10.1016/j.jclepro.2021.126153.

[36] L. Li, Y. He, H. Zhang, J.C.H. Fung, A.K.H. Lau, Enhancing IAQ, thermal comfort, and energy efficiency through an adaptive multi-objective particle swarm optimizer-grey wolf optimization algorithm for smart environmental control, Build. Environ. 235 (May 2023) 110235, https://doi.org/10.1016/j.buildenv.2023.110235.

[37] R. Shan, L. Junghans, Multi-objective optimization for high-performance building facade design: a systematic literature review, Sustainability 15 (21) (2023) 21, https://doi.org/10.3390/su152115596.

[38] H. Hosamo, S. Mazzetto, Data-driven ventilation and energy optimization in smart office buildings: insights from a high-resolution occupancy and indoor climate dataset, Sustainability 17 (1) (2025) 1, https://doi.org/10.3390/su17010058.

[39] M. Rabani, G.B.A. Coelho, A.J. Petersen, in: Enhancing Climate Resilience in Mixed-Mode Buildings: A Study of Hybrid Ventilation Strategies in a Cold Climate, Springer Nature Switzerland, Cham, 2025, pp. 991–1002, https://doi.org/10.1007/978-3-031-69626-8_83.

[40] L. Li, Z. Qi, Q. Ma, W. Gao, X. Wei, Evolving multi-objective optimization framework for early-stage building design: improving energy efficiency, daylighting, view quality, and thermal comfort, Build. Simul. 17 (11) (Nov. 2024) 2097–2123, https://doi.org/10.1007/s12273-024-1178-6.

[41] A. Azzi, M. Tabaa, B. Chegari, H. Hachimi, Balancing sustainability and comfort: a holistic study of building control strategies that meet the global standards for efficiency and thermal comfort, Sustainability 16 (5) (2024) 5, https://doi.org/10.3390/su16052154.

[42] C.S. Meena, A. Kumar, V.P. Singh, A. Ghosh, *Sustainable Technologies for Energy Efficient Buildings*, 1st Edition, CRC Press, Boca Raton, 2024.

[43] Z. Qiu, Q. Yong, J. Wang, L. Liao, B. Yu, A multi-objective optimization framework for performance-based building design considering the interplay between buildings and urban environments, Energy Convers. Manag. 315 (Sep. 2024) 118793, https://doi.org/10.1016/j.enconman.2024.118793.

[44] S. Márquez-Sánchez, et al., Enhancing building energy management: adaptive edge computing for optimized efficiency and inhabitant comfort, Electronics 12 (19) (2023) 19, https://doi.org/10.3390/electronics12194179.

[45] A. Abbas, et al., Challenges and opportunities in quantum optimization, Nat. Rev. Phys. 6 (12) (Dec. 2024) 718–735, https://doi.org/10.1038/s42254-024-00770-9.

[46] K. Blekos, et al., A review on Quantum Approximate Optimization Algorithm and its variants, Phys. Rep. 1068 (Jun. 2024) 1–66, https://doi.org/10.1016/j.physrep.2024.03.002.

[47] J. Tilly, et al., The Variational Quantum Eigensolver: a review of methods and best practices, Phys. Rep. 986 (Nov. 2022) 1–128, https://doi.org/10.1016/j.physrep.2022.08.003.

[48] S. Yarkoni, E. Raponi, T. Bäck, S. Schmitt, Quantum annealing for industry applications: introduction and review, Rep. Prog. Phys. 85 (10) (Sep. 2022) 104001, https://doi.org/10.1088/1361-6633/ac8c54.

[49] T. Li, Z.-Q. Yin, Quantum superposition, entanglement, and state teleportation of a microorganism on an electromechanical oscillator, Sci. Bull. 61 (2) (Jan. 2016) 163–171, https://doi.org/10.1007/s11434-015-0990-x.

[50] Y. Wang, C. Wei, Design optimization of office building envelope based on quantum genetic algorithm for energy conservation, J. Build. Eng. 35 (Mar. 2021) 102048, https://doi.org/10.1016/j.jobe.2020.102048.

[51] L. He, W. Wang, Design optimization of public building envelope based on multi-objective quantum genetic algorithm, J. Build. Eng. 91 (Aug. 2024) 109714, https://doi.org/10.1016/j.jobe.2024.109714.

[52] Z. Deng, X. Wang, B. Dong, Quantum computing for future real-time building HVAC controls, Appl. Energy 334 (Mar. 2023) 120621, https://doi.org/10.1016/j.apenergy.2022.120621.

[53] A. Ajagekar, T. Humble, F. You, Quantum computing based hybrid solution strategies for large-scale discrete-continuous optimization problems, Comput. Chem. Eng. 132 (Jan. 2020) 106630, https://doi.org/10.1016/j.compchemeng.2019.106630.

[54] An overview of recent advances', S. Golestan, M. R. Habibi, S. Y. Mousazadeh Mousavi, J. M. Guerrero, and J. C. Vasquez, 'Quantum computation in power systems, Energy Rep. 9 (Dec. 2023) 584–596, https://doi.org/10.1016/j.egyr.2022.11.185.

[55] D. Brockhoff, T. Friedrich, and F. Neumann, 'Analyzing Hypervolume Indicator Based Algorithms', in *Parallel Problem Solving from Nature – PPSN X*, G. Rudolph, T. Jansen, N. Beume, S. Lucas, and C. Poloni, Eds., Berlin, Heidelberg: Springer, 2008, pp. 651–660 https://doi.org/10.1007/978-3-540-87700-4_65.

[56] A. Auger, J. Bader, D. Brockhoff, E. Zitzler, Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications, Theor. Comput. Sci. 425 (Mar. 2012) 75–103, https://doi.org/10.1016/j.tcs.2011.03.012.

[57] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, 'Difficulties in specifying reference points to calculate the inverted generational distance for many-objective optimization problems', in *2014 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM)*, Dec. 2014, pp. 170–177 https://doi.org/10.1109/MCDM.2014.7007204.

[58] X. Cai, Y. Xiao, M. Li, H. Hu, H. Ishibuchi, X. Li, A Grid-based Inverted Generational Distance for Multi/Many-Objective Optimization, IEEE Trans. Evol. Comput. 25 (1) (Feb. 2021) 21–34, https://doi.org/10.1109/TEVC.2020.2991040.

[59] M.R. Sheldon, M.J. Fillyaw, W.D. Thompson, The use and interpretation of the Friedman test in the analysis of ordinal-scale data in repeated measures designs, Physiother. Res. Int. 1 (4) (1996) 221–228, https://doi.org/10.1002/pri.66.

[60] J. Ma, et al., A comprehensive comparison among metaheuristics (MHs) for geohazard modeling using machine learning: Insights from a case study of landslide displacement prediction, Eng. Appl. Artif. Intell. 114 (Sep. 2022) 105150, https://doi.org/10.1016/j.engappai.2022.105150.

[61] A. Di Meglio, et al., Quantum Computing for High-Energy Physics: State of the Art and challenges, PRX Quantum 5 (3) (Aug. 2024) 037001, https://doi.org/10.1103/PRXQuantum.5.037001.

[62] G. G. Guerreschi and M. Smelyanskiy, 'Practical optimization for hybrid quantum-classical algorithms', Jan. 05, 2017, *arXiv*: arXiv:1701.01450 https://doi.org/10.48550/arXiv.1701.01450.

[63] D.J. Egger, J. Mareček, S. Woerner, Warm-starting quantum optimization, Quantum 5 (Jun. 2021) 479, https://doi.org/10.22331/q-2021-06-17-479.

[64] E. Farhi, J. Goldstone, and S. Gutmann, 'A Quantum Approximate Optimization Algorithm', Nov. 2014 https://doi.org/10.48550/arXiv.1411.4028.

[65] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, M.D. Lukin, Quantum Approximate Optimization Algorithm: Performance, Mechanism, and Implementation on Near-Term Devices, Phys. Rev. X 10 (2) (Jun. 2020) 021067, https://doi.org/10.1103/PhysRevX.10.021067.

[66] J.R. McClean, J. Romero, R. Babbush, A. Aspuru-Guzik, The theory of variational hybrid quantum-classical algorithms, New J. Phys. 18 (2) (Feb. 2016) 023023, https://doi.org/10.1088/1367-2630/18/2/023023.

[67] U.R.S. Roy, S. Nayek, Optimization with Quantum Genetic Algorithm, Int. J. Comput. Appl. 102 (16) (Sep. 2014) 1–7.

[68] R. Shaydulin, I. Safro, and J. Larson, 'Multistart Methods for Quantum Approximate optimization', in *2019 IEEE High Performance Extreme Computing Conference (HPEC)*, Sep. 2019, pp. 1–8 https://doi.org/10.1109/HPEC.2019.8916288.

[69] X. Bonet-Monroig, et al., Performance comparison of optimization methods on variational quantum algorithms, Phys. Rev. A 107 (3) (Mar. 2023) 032407, https://doi.org/10.1103/PhysRevA.107.032407.

[70] S. Bravyi, D. Gosset, R. Movassagh, Classical algorithms for quantum mean values, Nat. Phys. 17 (3) (Mar. 2021) 337–341, https://doi.org/10.1038/s41567-020-01109-8.

[71] E. Campbell, A. Khurana, A. Montanaro, Applying quantum algorithms to constraint satisfaction problems, Quantum 3 (Jul. 2019) 167, https://doi.org/10.22331/q-2019-07-18-167.

[72] H. Goto, et al., High-performance combinatorial optimization based on classical mechanics, Sci. Adv. 7 (6) (2021) eabe7953, https://doi.org/10.1126/sciadv.abe7953.

[73] A. Bounceur, M. Hammoudeh, B. Adebisi, and A. Laouid, 'Grover's Algorithm: Quantum Brute-Force Search', in *Quantum Computing*, 1st Edition., Boca Raton: CRC Press, 2024.

[74] M. Monnet *et al.*, 'Pooling Techniques in Hybrid Quantum-Classical Convolutional Neural Networks', in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, Sep. 2023, pp. 601–610 https://doi.org/10.1109/QCE57702.2023.00074.

[75] 'Byggteknisk forskrift (TEK10)', Direktoratet for byggkvalitet. Accessed: Jan. 23, 2025. [Online]. Available: https://www.dibk.no/regelverk/tek.

[76] 'NS 3701:2012', Standard Online. Accessed: Jan. 23, 2025. [Online]. Available: https://online.standard.no/nb/ns-3701-2012.

[77] H. Hosamo, G.B.A. Coelho, C. Nordahl Rolfsen, D. Kraniotis, Building performance optimization through sensitivity analysis, and economic insights using AI, Energy Build. 325 (2024) 114999, https://doi.org/10.1016/j.enbuild.2024.114999.

[78] T. Chen and C. Guestrin, 'XGBoost: A Scalable Tree Boosting System', in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2016, pp. 785–794 https://doi.org/10.1145/2939672.2939785.

[79] A. Gupta, R. Joshi, N. Kanvinde, P. Gerela, and R. M. Laban, 'Metric Effects Based on Fluctuations in Values of k in Nearest Neighbor Regressor', in *Data Intelligence and Cognitive Informatics*, I. J. Jacob, S. Kolandapalayam Shanmugam, and I. Izonin, Eds., Singapore: Springer Nature, 2023, pp. 131–140 https://doi.org/10.1007/978-981-19-6004-8_12.

[80] F. Alzu'bi, A. Al-Rawabdeh, A. Almagbile, Predicting air quality using random forest: a case study in amman-zarqa, Egypt. J. Remote Sens. Space Sci. 27 (3) (2024) 604–613, https://doi.org/10.1016/j.ejrs.2024.07.004.

[81] R. Wang, S. Lu, W. Feng, A novel improved model for building energy consumption prediction based on model integration, Appl. Energy 262 (Mar. 2020) 114561, https://doi.org/10.1016/j.apenergy.2020.114561.

[82] Z. Ni, C. Zhang, M. Karlsson, S. Gong, A study of deep learning-based multi-horizon building energy forecasting, Energy Build. 303 (Jan. 2024) 113810, https://doi.org/10.1016/j.enbuild.2023.113810.

[83] 'What are hyperparameters? — Klu'. Accessed: Feb. 21, 2025. [Online]. Available: https://klu.ai/glossary/hyperparameters.

[84] C.A.C. Coello, G.T. Pulido, M.S. Lechuga, Handling multiple objectives with particle swarm optimization, IEEE Trans. Evol. Comput. 8 (3) (Jun. 2004) 256–279, https://doi.org/10.1109/TEVC.2004.826067.

[85] T. Kadowaki, H. Nishimori, Quantum annealing in the transverse Ising model, Phys. Rev. E 58 (5) (Nov. 1998) 5355–5363, https://doi.org/10.1103/PhysRevE.58.5355.

[86] 'D-Wave Ocean Software Documentation — Ocean Documentation 8.1.0 documentation'. Accessed: Jan. 23, 2025. [Online]. Available: https://docs.ocean.dwavesys.com/en/stable/.

[87] C. Yang, et al., How can SHAP (SHapley Additive exPlanations) interpretations improve deep learning based urban cellular automata model? Comput. Environ. Urban Syst. 111 (Jul. 2024) 102133 https://doi.org/10.1016/j.compenvurbsys.2024.102133.

[88] Y. Qiu, X. Yang, S. Chen, An improved gray wolf optimization algorithm solving to functional optimization and engineering design problems, Sci. Rep. 14 (1) (Jun. 2024) 14190, https://doi.org/10.1038/s41598-024-64526-2.

[89] R. H. Malonjao, J. B. Malonjao, and P. J. Borromeo, 'Friedman Chi-Square Test Analysis of Carbon Steel Tensile Stress Data Using Python', in *2024 IEEE International Conference on Technology, Informatics, Management, Engineering and Environment (TIME-E)*, Aug. 2024, pp. 44–50 https://doi.org/10.1109/TIME-E62724.2024.10919828.

[90] L. Štěpánek, F. Habarta, I. Malá, and L. Marek, 'A short note on post-hoc testing using random forests algorithm: Principles, asymptotic time complexity analysis, and beyond', in *2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS)*, Sep. 2022, pp. 489–497 https://doi.org/10.15439/2022F265.

[91] T. Lubinski, et al., Optimization applications as quantum performance benchmarks, ACM Trans. Quantum Comput. 5 (3) (2024) 18:1–18:44, https://doi.org/10.1145/3678184.

[92] J. Giraldo, J. Ossorio, N. M. Villegas, G. Tamura, and U. Stege, 'QPLEX: Realizing the Integration of Quantum Computing into Combinatorial Optimization Software', in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, Sep. 2023, pp. 1044–1049 https://doi.org/10.1109/QCE57702.2023.00118.

[93] Z. Deng and B. Dong, 'Discrete optimal control of building to grid integration based on quantum computing', presented at the Building Simulation 2023, in Building Simulation, vol. 18. IBPSA, 2023, pp. 2776–2783 https://doi.org/10.26868/25222708.2023.1545.

[94] J. Chae, C. A. Steed, and T. S. Humble, 'Visual Analytics of Performance of Quantum Computing Systems and Circuit Optimization', in *2024 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, Jul. 2024, pp. 613–618 https://doi.org/10.1109/ISVLSI61997.2024.00116.

[95] A. Abbas *et al.*, 'Quantum Optimization: Potential, Challenges, and the Path Forward', Los Alamos National Laboratory (LANL), Los Alamos, NM (United States), LA-UR-23-33327 https://doi.org/10.2172/2229681.