ORIGINAL ARTICLE

FastML-GA: FPGA-accelerated machine learning for real-time energy HVAC optimization in buildings

Mohammed Mshragi · Ioan Petri

Received: 17 July 2025 / Accepted: 8 November 2025

Abstract
Fast machine learning (FastML) has strong potential to enhance energy optimization and operational efficiency in heating, ventilation, and air conditioning (HVAC) systems within building management systems (BMS). Traditional HVAC control approaches frequently depend on static schedules and computationally intensive, CPU-based optimization techniques, which often lack the responsiveness and scalability required for real-time embedded applications. To address these limitations, we propose a fast machine learning framework that integrates a random forest surrogate model implemented as a hardware accelerator on the programmable logic (PL) with a lightweight and adaptive genetic algorithm (GA) executed on the processing system (PS), thereby forming a hybrid PS–PL deployment. This combination of fast machine learning and evolutionary algorithms optimization delivers substantial computational efficiency, achieving over 1.67 million predictions per second on a PYNQ-Z1 FPGA and significantly outperforming recent FPGA-based approaches. By using a case study, we demonstrate how FastML can employ a GA multi-objective fitness function to dynamically optimize hourly airflow rates and supply air temperatures in response to occupancy and seasonal environmental patterns, thereby reducing electricity and thermal energy consumption while maintaining occupant comfort within standard predicted mean vote (PMV) thresholds. Empirical evaluation conducted over 72 days across four distinct seasons reveals consistent electricity savings exceeding 50%, alongside thermal energy reductions of up to 150 kWh per day during heating periods. A comprehensive three-dimensional Pareto front analysis further substantiates the system's capability to effectively balance energy efficiency and occupant comfort. These results highlight the practicality, scalability, and substantial promise of FPGA-based multi-objective optimization as a robust, real-time solution for intelligent and sustainable building energy management at the edge.

Keywords Fast machine learning · Genetic algorithms · Building management systems · Energy efficiency · Optimisation

Abbreviations

AFR	Air flow rate
AT	Air temperature
AXI	Advanced eXtensible interface
BMS	Building management system
BRAM	Block random-access memory

Published online: 28 January 2026

CoP	Coefficient of performance
CPU	Central processing unit
DDPG	Deep deterministic policy gradient
DL	Deep learning
DMA	Direct memory access
DRL	Deep reinforcement learning
DSP	Digital signal processing (block)
FastML-GA	Fast machine learning with genetic algorithm
FPGA	Field-programmable gate array
GA	Genetic algorithm
GA-Opt	Genetic algorithm optimization
HLS	High-level synthesis
HVAC	Heating, ventilation, and air conditioning
II	Initiation interval
IoT	Internet of Things
MAE	Mean absolute error
ML	Machine learning
MPC	Model predictive control
PMV	Predicted mean vote
RBFNN	Radial basis function neural network
RF	Random forest
RFR	Random forest regressor
RH	Relative humidity
RMSE	Root mean square error
RT	Room temperature
RTL	Register transfer level
SEMS	Smart energy management system
SoC	System on chip
WT	Water temperature

1 Introduction

The building sector is a cornerstone of global decarbonisation strategies, accounting for approximately 34% of global final energy consumption and 37% of energy-related CO_2 emissions as of 2022 [1]. Heating, ventilation, and air conditioning (HVAC) systems, which can consume up to 40% of a building's total energy [2], represent a critical opportunity for improving energy efficiency and reducing environmental impact. Optimizing HVAC operations is essential not only for achieving climate goals but also for enhancing occupant comfort and supporting the long-term sustainability of the built environment.

Traditional building automation systems (BAS) rely heavily on rule-based or schedule-driven control strategies, which are often inadequate for responding to dynamic factors such as fluctuating occupancy, real-time weather changes, and seasonal variations [3, 4]. These static approaches frequently lead to energy waste, suboptimal thermal comfort, and failure to meet stringent energy efficiency targets. For instance, fixed schedules may maintain HVAC operation during unoccupied periods, exacerbating energy consumption due to the thermal inertia of buildings, which causes temperature changes to lag behind occupancy shifts [5]. Such inefficiencies underscore the need for adaptive, intelligent control systems capable of real-time decision-making.

Recent advancements in data-driven predictive control have shown promise in addressing these challenges. By leveraging real-time sensor data—such as occupancy, temperature, and humidity—and historical building performance records, machine learning (ML) models can accurately forecast indoor climate dynamics and dynamically adjust HVAC setpoints [6]. Among ML forecasting techniques, random forests (RF) stand out for their robustness to noisy inputs, ability to model complex nonlinear relationships, and low inference complexity, making them ideal for deployment on resource-constrained edge devices [7, 8]. RF models serve as effective surrogate models, predicting key performance metrics like electricity consumption, thermal energy use, and thermal comfort (via the Predicted Mean Vote, PMV) with high accuracy.

However, most ML-based HVAC systems rely on conventional CPU-based or cloud-computing architectures, which introduce significant limitations for real-time applications. Cloud-based systems suffer from communication latencies due to round-trip data transfers, while CPU-based edge platforms, such as Raspberry Pi, often lack the computational power for advanced tasks like real-time video processing or high-throughput ML inference [9]. These delays can result in overheating, overcooling, or delayed ventilation, compromising both energy efficiency and occupant comfort. For example, even a few seconds of latency in processing occupancy signals can lead to prolonged HVAC operation, wasting energy in unoccupied spaces.

Field-programmable gate arrays (FPGAs) offer a compelling solution to these challenges. FPGAs provide ultra-low-latency inference and high energy efficiency, consuming significantly less dynamic power than general-purpose CPUs. Studies have shown that FPGA-accelerated workloads can reduce operational carbon footprints by up to 40% compared to CPU-based systems, making them a sustainable choice for smart building applications [10]. By enabling task-specific hardware acceleration, FPGAs support rapid evaluation of thousands of candidate control strategies, a critical requirement for real-time HVAC optimization in dynamic environments.

The thermal mass of buildings further complicates HVAC control, as temperature changes occur gradually rather than instantaneously [5]. This inertia can lead to unnecessary heating or cooling after occupants leave, increasing energy consumption. Predictive control models, pre-trained to account for building thermal dynamics, can anticipate such changes and adjust HVAC setpoints proactively. Studies demonstrate that predictive approaches can achieve energy savings ranging from 1% to 13.3%, depending on the balance between thermal comfort and efficiency [11]. Fast machine learning (FastML), characterized by sub-millisecond inference times, is thus essential for enabling real-time, localized decision-making at the edge, where environmental and occupancy signals can change within seconds.

Evolutionary algorithms, particularly Genetic algorithms (GAs), offer significant potential for multi-objective HVAC optimization by exploring complex setpoint spaces to balance energy use, comfort, and operational stability in buildings [12]. However, their integration with ML models on embedded platforms has been limited by computational constraints and latency issues. Most GA implementations rely on high-performance computing resources, which are impractical for edge deployment in resource-constrained environments such as smart buildings.

This work introduces `FastML-GA`, a novel FPGA-accelerated framework that integrates a random forest (RF) surrogate model implemented as a hardware accelerator on the programmable logic (PL) with a lightweight genetic algorithm (GA) executed on the embedded ARM Cortex-A9 CPUs of the PYNQ-Z1 processing system (PS), forming a hybrid PS–PL architecture for real-time HVAC optimization in buildings. FastML-GA achieves over 1.67 million predictions per second, surpassing existing FPGA-based energy management systems [13]. By combining the low-latency inference of the FPGA-based RF with the optimization capabilities of the GA running on the PS, the framework enables rapid evaluation of thousands of candidate setpoints, ensuring timely responses to dynamic building conditions. Unlike traditional CPU- or cloud-based solutions, FastML-GA delivers high-throughput, energy-efficient control suitable for real-time applications.

We validate FastML-GA through seasonal simulations using high-fidelity building datasets, achieving over 50% reductions in electricity and thermal energy consumption while maintaining thermal comfort within ASHRAE Standard 55-2020 guidelines (PMV index) [14, 15]. Pareto analysis demonstrates the framework's ability to navigate trade-offs between energy efficiency and comfort across diverse operational scenarios, highlighting its

adaptability to varying occupancy and environmental conditions. Additionally, the system's FPGA-based implementation minimizes operational emissions, aligning with broader sustainability goals for smart buildings.

By advancing the integration of FastML and embedded optimization, FastML-GA contributes to the decarbonization of the building sector and the development of responsive, sustainable building management systems. The rest of the paper is structured as follows: Sect. 2 reviews related literature, Sect. 3 details the FastML-GA framework, Sect. 4 describes the experimental setup, Sect. 5 presents results and analysis, and Section 6 concludes with final remarks and future research directions.

2 Related work

Improving building energy performance is central to the global effort to decarbonize the built environment, with intelligent HVAC control representing a critical opportunity for energy transition and sustainability. The integration of machine learning (ML) into building energy systems has gained increasing attention as a means to reduce operational emissions and meet net-zero targets. A recent special issue edited by Guo et al. [16] consolidates major developments in data-driven HVAC control, highlighting the role of ML in performance forecasting, fault detection, and pattern recognition—tools critical for advancing low-energy building operations.

Numerous studies have addressed the computational burden of simulation-based HVAC optimization by replacing costly simulations with ML-based surrogate models. For example, Araújo et al. [17] combined a validated building energy simulation tool with ML approximators, achieving up to 22% cost savings and 100× faster execution. However, such methods typically remain confined to CPU-bound pipelines and lack support for embedded or real-time deployment, limiting their practical utility in dynamic control environments.

A comprehensive review by Ala'raj et al. [18] classified data-driven HVAC optimization strategies and highlighted the effectiveness of models such as random forests and neural networks for predicting thermal comfort and energy use. These findings were reinforced by Wang et al. [19], who applied a random forest model for hourly energy prediction in educational buildings, outperforming Regression Trees and SVR by 14–25% and 5-−5.5%, respectively. Their study also showed that feature importance varies with seasons, underscoring the adaptability of RFs to dynamic conditions and climatic variability—factors crucial for low-carbon HVAC design.

ML models such as artificial neural networks (ANNs), support vector machines (SVMs), and random forests (RFs) have also been widely combined with evolutionary optimization methods to balance the competing objectives of energy reduction and occupant comfort. Ilbeigi et al. [20] proposed a multi-layer perceptron (MLP) trained on EnergyPlus simulation data and optimized via the Galapagos genetic algorithm, achieving 35% energy savings. Similarly, Ferreira et al. [21] demonstrated that ANN-based predictive control could yield over 50% reductions in energy use, highlighting their potential for real-world building decarbonisation.

Other recent innovations in modeling include symbolic regression [22], which achieved a 16.1% reduction in peak power using a model predictive control framework, and reinforcement learning (RL)-based HVAC control. For instance, Ding et al. [23] proposed CLUE, a model-based RL system using Gaussian Processes for safe and data-efficient HVAC operation, while [2] applied deep RL in multi-zone office buildings, achieving up to 37% energy savings.

The design of multi-energy systems for net/nearly zero energy buildings (NZEBs) has also advanced, with recent work by Lu et al. [24] addressing correlated uncertainties in system sizing using a copula-based scenario generator and NSGA-II optimization. Their results demonstrate high energy self-sufficiency and thermal comfort, reinforcing the need for uncertainty-aware, multi-objective optimization in smart buildings, a challenge we address from a real-time, embedded perspective. Evolutionary algorithms (EAs) remain a popular choice for multi-objective HVAC optimization. However, most implementations operate offline, are simulation-heavy, and lack real-time applicability or embedded deployment, thus limiting their practical integration into energy-responsive building management systems.

In contrast to these approaches, our work eliminates runtime simulation entirely. We train a random forest regressor (RFR) on pre-generated HVAC simulation data and deploy it on FPGA hardware for high-throughput, low-latency inference. This is coupled with a lightweight, multi-objective genetic algorithm tailored for HVAC control, enabling dynamic, scenario-based optimization directly at the edge. Our system achieves electricity savings exceeding 50% and up to 23.95× performance gains over CPU-based methods, while maintaining high prediction accuracy and control robustness.

Furthermore, FPGA-based edge computing is inherently more energy-efficient compared to CPU- or GPU-based systems. For instance, Xu et al. [25] demonstrated that FPGA edge deployment for mobile vision tasks reduced response time by 3×, execution time by 15×, and power consumption across both mobile devices and edge nodes. More broadly, Vaithianathan et al. [26] showed that low-power FPGA techniques such as dynamic voltage scaling and clock gating can achieve over 40% energy savings in mobile and embedded applications. In addition, configuration-aware and duty-cycling strategies for FPGA deep-learning accelerators can cut configuration energy by $\sim 40\times$ and extend operational lifetime via idle-waiting and adaptive strategy switching [27]. This is particularly important for reducing the carbon footprint of edge devices, which often operate continuously and in large numbers. To the best of our knowledge, no prior work has combined ML-based surrogate modeling and genetic algorithm optimization for HVAC control on FPGA hardware. Although GA and ML accelerators have been independently explored in other domains, our work is the first to unify both into a compact, embedded, and energy-efficient optimization framework.

Beyond HVAC-specific studies, recent work on embedded AI highlights the portability of ML frameworks across microcontrollers (MCUs), FPGAs, and GPUs. TinyML benchmarks demonstrate that modern MCUs such as the STM32N6 can achieve inference latencies of 1.6–2.9 ms with energy consumption as low as 153–331 μJ per inference, making them attractive for ultra-low-power but higher-latency applications [28]. Lightweight FPGA boards such as the PYNQ-Z1 have been shown to deliver up to 4.54×10^5 inferences/s at only 1.87 W [29], while higher-capacity platforms like the Xilinx ZCU102 MPSoC further reduce DNN latency and surpass GPU baselines in energy efficiency [30, 31]. Conversely, Nvidia Jetson Orin GPUs achieve sub-millisecond inference and throughputs approaching 1 M inferences/s [32, 33], but at 20–60 W power consumption depending on mode [34]. These findings underline a clear trade-off: MCUs maximize energy efficiency, FPGAs strike the best balance between latency and power, and GPUs provide the highest throughput at significantly higher energy cost. Our framework builds on these insights by targeting FPGAs as a practical compromise for real-time HVAC optimization, while remaining portable to MCUs or GPUs depending on deployment constraints.

By tightly integrating a random forest surrogate with a season-adaptive genetic algorithm on FPGA, our system delivers sub-millisecond inference latency, ensures thermal comfort compliance, and supports scalable optimization under dynamic environmental and occupancy conditions. This represents a novel and practical advancement in intelligent building control—contributing directly to the goal of decarbonizing the built environment through embedded, high-performance HVAC optimization.

While ML-based surrogate models and genetic algorithms (GAs) have been individually applied in building energy and HVAC optimization, they have rarely been integrated into a unified, embedded framework, particularly not within a FastML-system that ensures real-time energy optimisation capability. As summarized in Table 1, recent studies typically focus on either ML or GA—often relying on CPU-based simulations—and lack real-time or FPGA-based implementations. This gap underscores the novelty of our approach, which unifies ML-based surrogate modeling and GA-based optimization within an energy-efficient FPGA framework for real-time HVAC optimisation in buildings.

Table 1 Related studies on ML, FPGA, and GA in building energy management and HVAC

References	Research direction	ML/AI method	FPGA/GA	Application	Key contr
[13]	Fast ML for BMS FPGA	LSTM	FPGA	Energy prediction	520+ inf/s, low-latency
[35]	HVAC control via ANN-MPC	ANN + MPC	None	Energy saving	27–39% energy saved
[36]	Smart EMS on FPGA	None	FPGA	Demand response	Grid–battery switching
[37]	RBFNN + MPC control	RBFNN + MPC	None	HVAC efficiency	15% saved, validated
[38]	DRL for HVAC control	DRL (multi-agent)	None	HVAC cost saving	75% saved, scalable
[39]	Comfort–energy optimization	DDPG + DNN	None	HVAC control	Comfort–energy trade-off
[40]	Review of ML in BMS	SVM, RF, CNN, RL	None	Faults, prediction	Hybrid ML, explainability
[41]	Temp forecasting (multi-zone)	LSTM (seq2seq)	None	Temp prediction	Seq2seq, uncertainty-aware
[42]	AI anomaly detection review	DL, RF, Clustering	None	Fault detection	Dataset, reproducibility
[43]	GA-tuned PID on FPGA	None	FPGA + GA	PID control	Real-time PID tuning
[44]	GA vs PSO for HVAC	None	GA	Setpoint optimization	MOGA vs NSGA-II/III
[45]	GA-based model validation	None	GA	Fault localization	Detects mismatches
[46]	GA on FPGA (parallel)	None	FPGA + GA	GA tasks	170,000× speedup
[47]	FPGA-based SEMS for microgrid	None	FPGA	Load management	Fast SEMS

3 Methodology

Real-time HVAC control in buildings requires immediate response to fluctuating energy signals such as occupancy, weather, and internal load changes. Delays in inference or actuation can lead to energy waste or occupant discomfort. Conventional CPU- or cloud-based ML pipelines often introduce latency that is incompatible with these dynamic conditions. To address this, we integrate fast machine learning (FastML) into our energy optimisation framework. Implemented via a random forest regressor on FPGA hardware, FastML delivers sub-millisecond inference latency and high throughput—enabling the evaluation of thousands of candidate setpoints generated by the genetic algorithm in real time. This allows prompt selection of optimal control actions, ensuring timely and energy-efficient HVAC operation. By supporting ultra-fast, localized decision-making, FastML enhances the responsiveness of smart buildings and directly contributes to emissions reduction and sustainability goals.

3.1 Overview of the FastML-GA framework

The proposed methodology combines fast machine learning with genetic algorithms to optimize energy consumption in buildings by determining HVAC setpoints in real time. The overall system architecture is depicted in Fig. 1, and is composed of three key components:

1. *Surrogate model module (Random forest regressor)*: A trained multi-output random forest regressor is used to approximate the building performance metrics (e.g., electricity consumption, thermal energy use, and thermal comfort) for a given set of HVAC setpoints.
2. *Hardware acceleration module (HLS and FPGA)*: The trained model is exported to synthesizable C++ using High-Level Synthesis (HLS) and deployed on FPGA hardware. This enables low-latency, high-throughput inference via batched AXI-Stream interfaces.
3. *Genetic algorithm (GA) optimization module*: A lightweight GA searches the HVAC setpoint space to minimize energy usage while maintaining thermal comfort. Candidate solutions are evaluated in batches using the FPGA-accelerated surrogate model.

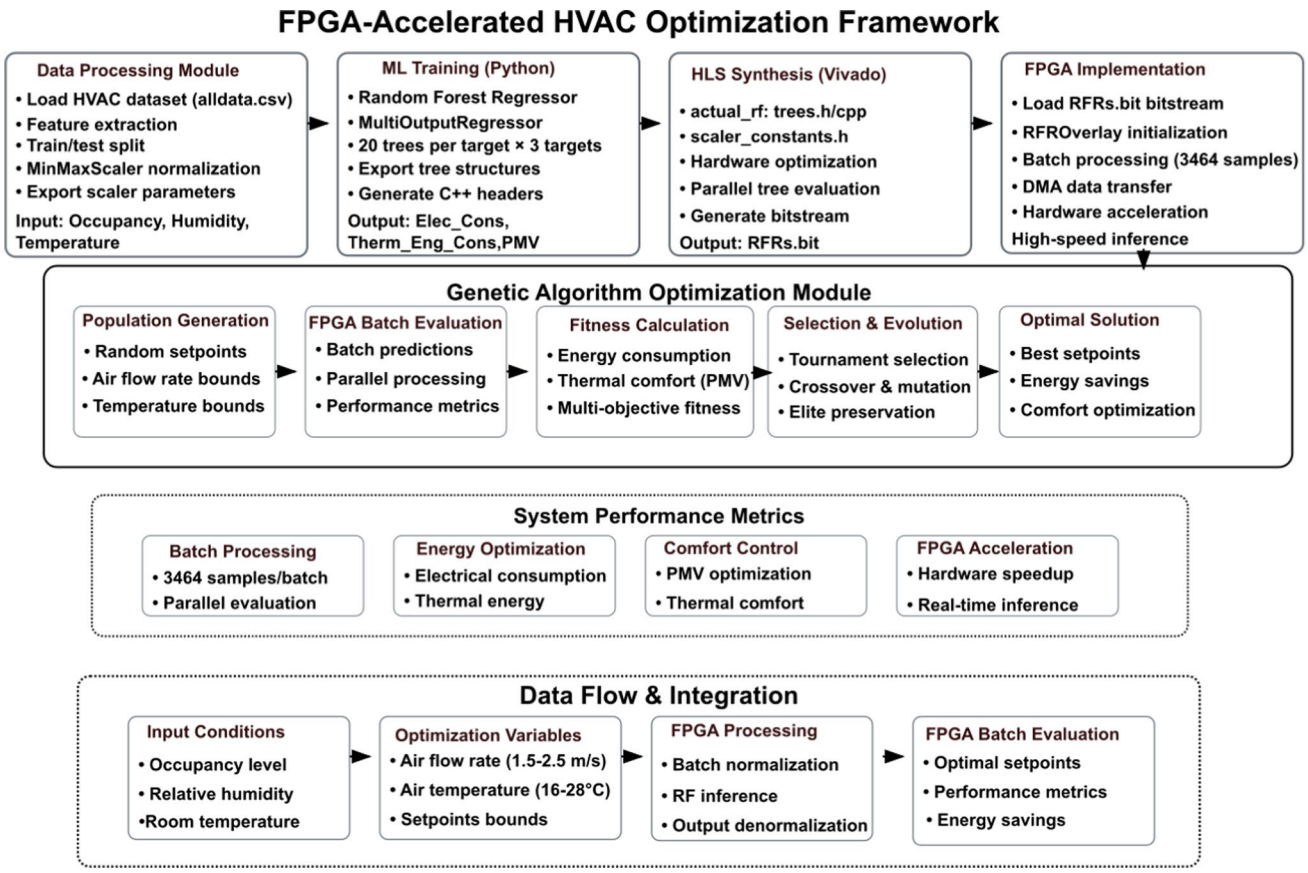

Fig. 1 System architecture illustrating the FPGA-accelerated HVAC optimization framework. A random forest-based surrogate model is deployed on the FPGA to enable high-throughput batch evaluation of candidate solutions generated by the genetic algorithm

Table 2 FIDIA sports facility characteristics

Attribute	Description
Location	Rome, Italy
Building structure	Wooden external walls and roof (9 cm thick), concrete floor, single-glass windows (thermal transmittance: 5.7 W/m^{2}K, solar gain: 0.7)
Geometry	Gable roof ($H_{\min}$ = 3 m, $H_{\max}$ = 6 m), window surfaces ≈ 70 m^2
Facilities	Indoor pool (25 m × 16 m, 760 m^3), learning pool (16 m × 4 m, 64 m^3), gym, fitness room (486 m^3), volleyball court (8960 m^3), outdoor tennis and five-a-side courts
Metering	Electricity, thermal, and water metering; co-generation units; solar thermal collectors; gas boilers; PMV sensors; 1-h sampling
Monitoring period	72 days, covering occupancy, temperature, humidity, and energy consumption during operational hours (Monday–Friday 08:00–22:00, Saturday 08:00–18:00, Sunday 10:00–13:00)

3.2 Random forest surrogate model for HVAC prediction

This module operates in coordination with the data processing and genetic algorithm (GA) modules. It learns the mapping from HVAC control and environmental inputs to energy and comfort outputs, acting as a fast and reliable surrogate model for optimization.

This study utilizes a dataset from the FIDIA sports facility in Rome, Italy, described by [48]. The facility features wooden external walls and roof (9 cm thick), swimming pools, gymnasiums, and multi-purpose courts. It employs cogeneration units, solar thermal collectors, gas boilers, and advanced metering systems (Table 2).

Table 3 PMV thermal comfort scale

PMV value	Thermal sensation
+3	Hot
+2	Warm
+1	Slightly warm
0	Neutral
-1	Slightly cool
-2	Cool
-3	Cold

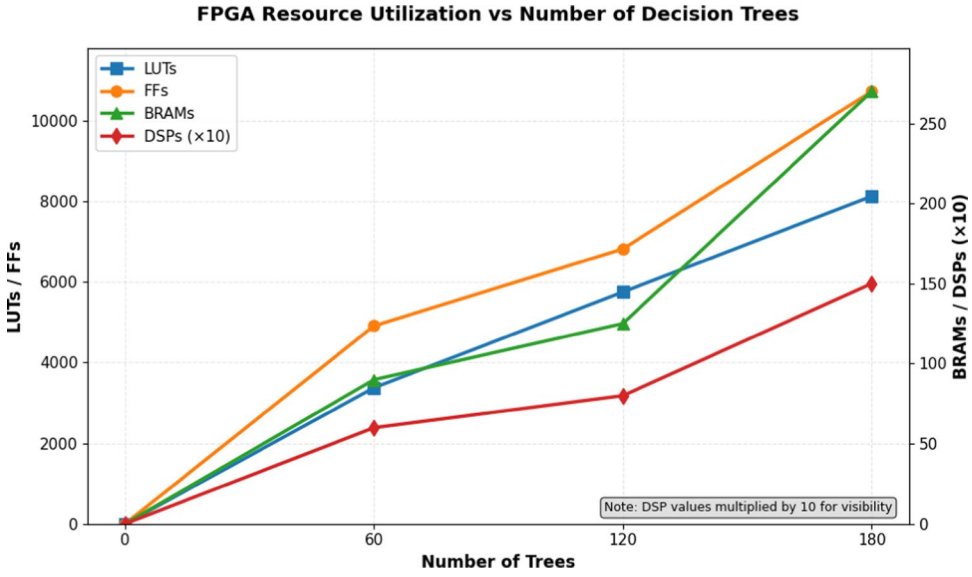

Fig. 2 Estimated FPGA resource utilization for different numbers of decision trees in a 3-output random forest model. LUTs and FFs are shown on the left y-axis, while BRAMs and DSPs are plotted on the right y-axis. Resource usage scales linearly with model complexity

The dataset, generated through 72 days of high-fidelity HVAC simulation, contains approximately 4320 hourly-aggregated entries.

The RFR model predicts electricity consumption in kWh from HVAC fans and compressors($Elec\_Cons$), thermal energy consumption in kWh for heating and cooling loads ($Therm\_Eng\_Cons$), and PMV (Predicted Mean Vote) from environmental and control inputs, while the GA module optimizes HVAC settings to minimize energy consumption and maintain thermal comfort. The PMV metric quantifies thermal comfort on a 7-point scale ranging from -3 (cold) to $+3$ (hot), as shown in Table 3, and is widely adopted in HVAC research [49].

Data preprocessing included removing incomplete entries, normalizing features and targets using the `MinMaxScaler` from the `scikit-learn` Python package, and splitting the dataset into 80% training and 20% testing subsets. Output targets were jointly scaled to preserve their relative magnitudes.

The RFR was selected for its robustness to overfitting, capacity to model both linear and nonlinear relationships, and native support for multi-output regression via the MultiOutputRegressor wrapper [50]. Each target output is predicted by an ensemble of 20 decision trees with a maximum depth of 6, totaling 60 trees. This configuration balances predictive accuracy with hardware feasibility on FPGA.

Empirical tests showed that model accuracy plateaued beyond 20 trees per output and depth 6, with R^2 and RMSE metrics stabilizing. Unlike neural networks that typically require quantization for speedup, random forests use ensemble averaging and are inherently quantization-free [51]. FPGA synthesis confirmed a linear growth in LUTs, flip-flops, BRAMs, and DSP usage with tree count, validating the 60-tree configuration (Fig. 2).

The trained RFR model was evaluated using Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and the R^2 coefficient of determination. Results showed strong alignment between predictions and actual values (Table 4), with high throughput suitable for real-time applications.

Table 4 Performance metrics of the RFR surrogate model

Target variable	RMSE	MAE	R^2
Elec_Cons	0.0027	0.0004	0.9826
Therm_Eng_Cons	1.5683	0.9280	0.9714
PMV	0.2330	0.1791	0.9673

RMSE and MAE are reported in kWh for Elec_Cons and Therm_Eng_Cons, and in PMV units for thermal comfort (PMV)

Linear regression was evaluated as a baseline model; however, it could not adequately capture the nonlinear interactions among HVAC variables such as airflow, supply air temperature, occupancy, and external conditions [52, 53]. Moreover, the wide disparity in target variable magnitudes (e.g., $Elec\_Cons$ and PMV in the range $[-6.4, 2.7]$ versus $Therm\_Eng\_Cons$ exceeding 39) introduced fixed-point scaling challenges during FPGA deployment, reducing accuracy when using a unified representation [54, 55]. Artificial Neural Networks (ANNs) were also considered, but their reliance on large numbers of multiplications and additions requires extensive DSP slices and LUT resources, making them far less efficient for lightweight FPGA platforms without aggressive quantization or pruning [56]. By contrast, random forests perform only threshold comparisons, minimizing DSP usage while remaining robust to seasonal variability and sensor noise [57]. This robustness is further supported by recent reviews highlighting sensor reliability and deployment challenges in building control systems [58, 59]. For these reasons, random forest was selected as the surrogate model.

For FPGA deployment, all trained decision trees were exported into fixed-format C++ header and source files. Each tree was encoded into a custom data structure using static arrays for node features, thresholds, child indices, leaf scores, and decision flags. Arrays were padded to 128 nodes to enable uniform indexing and avoid out-of-bounds behavior during traversal in C++, as required by Vivado HLS. Trees were grouped into a structure of `target_trees[N_TARGETS][N_TREES_PER_TARGET]`. Normalization constants were exported to enable real-time scaling on hardware.

The preprocessed test data was exported as C++ header files for direct simulation and synthesis use. Prediction timing was benchmarked on CPU for comparison. Three NumPy arrays—scaled test inputs, RFR predictions, and ground-truth labels—were saved for validating inference accuracy between Python and FPGA implementations.

3.2.1 HLS and hardware-accelerated implementation

We implemented a hardware-accelerated random forest regressor (RFR) on a PYNQ-Z1 FPGA to enable real-time HVAC optimization. As discussed in Sect. 3.2, simpler alternatives such as linear regression and ANNs were considered; however, linear regression could not capture the nonlinear HVAC dynamics effectively, while ANN models posed significant resource demands and required quantization and pruning, which may reduce model accuracy on FPGA deployment. Consequently, random forest was selected as a more practical balance between accuracy, robustness, and resource efficiency. The model predicts electricity consumption (kWh), thermal energy consumption (kWh), and Predicted Mean Vote (PMV) for thermal comfort assessment. The RFR was trained using `scikit-learn` in Python and subsequently exported to synthesizable C++ code using Vivado High-Level Synthesis (HLS). For deployment, we mapped 20 decision trees per output to hardware, totaling 60 trees, which achieves an optimal balance between inference accuracy and FPGA resource efficiency.

During model export and deployment, we encountered significant challenges due to the differing scales of the target variables: `Elec_Cons`, `Therm_Eng_Cons`, and `PMV`. Specifically, `Elec_Cons` and `PMV` exhibit small magnitudes, with `PMV` ranging from -6.4 to $+2.7$, while `Therm_Eng_Cons` can exceed 39. This disparity complicated the selection of a unified fixed-point format that could accommodate all targets with sufficient precision and dynamic range.

Initially, we adopted a fixed-point representation with 18-bit width and 8-bit integer portion for features and thresholds (`ap_fixed<18,8>`, where the notation `ap_fixed<Total,Int>` indicates a fixed-point number with `Total` total bits and `Int` integer bits, with the remaining bits allocated to fractional precision, and

18-bit width with 12-bit integer portion for scores (`ap_fixed<18,12>`). While this format provided the higher fractional resolution necessary for accurately representing `Therm_Eng_Cons`, it resulted in increased FPGA resource consumption.

To address this limitation, we applied Min-Max normalization to both input features and output targets during training, and implemented the corresponding normalization process in hardware. This transformation aligned the ranges of all targets, enabling the adoption of a more resource-efficient configuration with 16-bit width and 6-bit integer portion (`ap_fixed<16,6>`) for all data types. This reduced-precision setup preserved high prediction accuracy across all outputs, including the previously problematic `Therm_Eng_Cons`, while significantly decreasing hardware resource utilization.

All decision trees exported from Python were preprocessed with Min-Max normalization, and hardware modules were designed to apply normalization to input features before inference and to de-normalize outputs after prediction. This approach ensured full compatibility with fixed-point arithmetic on FPGA while maintaining high inference fidelity. For targets with larger dynamic ranges—such as `Therm_Eng_Cons`–this precise normalization was critical to avoid quantization errors. The normalized values allow the 16-bit fixed-point representation to retain sufficient fractional precision and dynamic range for real-time embedded inference.

The normalization and de-normalization processes are mathematically defined as:

$$X_{\text{norm}} = \frac{X - X_{\min}}{X_{\text{range}}} \qquad Y = Y_{\text{norm}} \cdot Y_{\text{range}} + Y_{\min} \tag{1}$$

The scaling constants $X_{\min}$, X_{range}, $Y_{\min}$, and Y_{range} were extracted from the `MinMaxScaler` and stored in a hardware-accessible header file. The specific values are:

$$X_{\min} = [0.0, 21.5072, -3.9189, 1.5, 4.8266] \tag{2}$$

$$X_{\text{range}} = [11.0, 78.4928, 43.9189, 2.2842, 30.0955] \tag{3}$$

$$Y_{\min} = [0.0349, 0.01, -6.4031] \tag{4}$$

$$Y_{\text{range}} = [0.0547, 39.4573, 9.1423] \tag{5}$$

The complete hardware inference pipeline is optimized using several HLS directives and modular design strategies. Each decision tree structure—comprising feature indices, thresholds, child node indices, and leaf values—is exported into static C++ arrays, which are used to reconstruct the model on FPGA.

During Vivado HLS synthesis, the core inference function is optimized using the following directives: (i) `HLS PIPELINE II=1` enables full pipelining to process one input sample per clock cycle; (ii) `HLS ARRAY_PARTITION complete` ensures simultaneous access to all tree nodes and feature arrays for parallel tree evaluation; (iii) `HLS UNROLL` unrolls tree evaluation loops to exploit the parallelism inherent in the FPGA fabric; (iv) `HLS RESOURCE core=AddSub_DSP` maps arithmetic operations to DSP slices, reducing LUT usage and improving timing; and (v) `HLS DATAFLOW` enables concurrent execution of different pipeline stages for throughput maximization.

To support high-speed data transfers, the system employs AXI-Stream interfaces combined with Direct Memory Access (DMA). This configuration eliminates CPU bottlenecks during inference and enables high-throughput batch predictions. Unlike AXI-Lite, which is suitable only for low-frequency transactions, AXI-Stream provides sufficient bandwidth to handle thousands of samples efficiently.

The tree traversal algorithm is implemented using a single-pass, loop-based structure to ensure bounded latency and avoid recursion. The procedure is detailed in Algorithm 1:

Algorithm 1 Hardware-optimized tree traversal

Require: Normalized features `features[N_FEATURES]`, tree index `tree_idx`, target index `target_idx`.
Ensure: Predicted score.

```
currentNode ← 0
for i = 1 to MAX_TREE_DEPTH do
    if isLeaf(currentNode, tree_idx, target_idx) then
        return getLeafValue(currentNode, tree_idx, target_idx)
    end if
    feature_idx ← getFeatureIndex(currentNode, tree_idx, target_idx)
    threshold ← getThreshold(currentNode, tree_idx, target_idx)
    if features[feature_idx] ≤ threshold then
        currentNode ← getLeftChild(currentNode)
    else
        currentNode ← getRightChild(currentNode)
    end if
end for
return getLeafValue(currentNode, tree_idx, target_idx)
```

Table 5 summarizes the FPGA resource utilization after synthesis. The efficient use of normalization and fixed-point arithmetic enabled substantial resource savings, allowing the model to be deployed on low-power FPGAs with sufficient headroom for additional logic.

This FPGA-based implementation of random forest regression delivers sub-millisecond inference latency and high throughput, supporting accurate prediction of HVAC control targets in real-time with limited hardware resources. The model's combination of robustness, efficiency, and low latency makes it ideally suited as a surrogate for genetic optimization in energy-efficient HVAC systems. The successful deployment demonstrates the viability of machine learning acceleration on resource-constrained embedded platforms for real-time building automation applications.

3.3 Genetic algorithm optimization

A custom seasonal genetic algorithm (GA) was developed to optimize the setpoints of a Heating, ventilation, and air conditioning (HVAC) system, leveraging predictions from a random forest (RF) surrogate model accelerated by a field-programmable gate array (FPGA). The FPGA enhances the speed of RF predictions, while the GA runs on the embedded ARM Cortex-A9 CPUs of the PYNQ-Z1 processing system (PS), determining optimal hourly air flow rates and temperatures across seasons. The optimization targets operational hours from 08:00 to 17:00 ($H = 10$ h), adapting to seasonal energy patterns and comfort requirements for heating and cooling in commercial buildings. The optimization process is illustrated in Fig. 3, which outlines the workflow from data initialization to result generation, highlighting FPGA setup, seasonal optimization, and GA operations.

Table 5 FPGA resource utilization summary

Resource type	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	–	3	–	–	–
Expression	–	–	0	1108	–
FIFO	0	–	40	240	–
Instance	–	3	0	270	–
Memory	90	–	0	0	–
Multiplexer	–	–	–	316	–
Register	0	–	4868	1440	–
Total	90	6	4908	3374	0
Available	280	220	106,400	53,200	0
Utilization (%)	32.1	2.7	4.6	6.3	0

3.3.1 Genetic algorithm optimization for FPGA HVAC

The optimization is formulated as a constrained, multi-objective minimization task, balancing electrical consumption, thermal energy (heating or cooling), thermal comfort, and setpoint smoothness. The decision variables form a vector

$$x = \{f_1, t_1, f_2, t_2, \ldots, f_H, t_H\} \in \mathbb{R}^{2H}$$

where f_i and t_i are the air flow rate (m^3/s) and air temperature (°C) for hour *i*, respectively. The season-specific objective function is:

$$\begin{aligned} \min_x \quad & f_s(x) = w_{E,s} \sum_{i=1}^{H} E_i + w_{T,s} \sum_{i=1}^{H} T_i^s + 100 \sum_{i=1}^{H} \phi(PMV_i) + 5S(x) \\ & - \delta_{s,\text{summer}} \cdot 0.1 \sum_{i=1}^{H} \max(30 - t_i, 0) \\ \text{s.t.} \quad & f_i \in [f_{\min,s}, f_{\max,s}], \quad t_i \in [t_{\min,s}, t_{\max,s}] \quad \text{if occupied} \\ & f_i \in [f_{\min,s} - 0.3, f_{\min,s}], \quad t_i \in [t_{\min,s} - 2, t_{\min,s}] \quad \text{if unoccupied} \end{aligned} \tag{6}$$

where E_i is electrical consumption (kWh), T_i^s is season-specific thermal energy (kWh), $\phi(PMV_i)$ is the comfort penalty, $S(x)$ is the smoothness penalty, and $\delta_{s,\text{summer}}$ is 1 for summer and 0 otherwise. The weights are:

$$w_{E,s}, w_{T,s} = \begin{cases} 0.7, 0.3 & \text{(summer)} \\ 0.4, 0.6 & \text{(winter)} \\ 0.5, 0.5 & \text{(spring/autumn)} \end{cases}$$

The summer efficiency bonus reduces cooling load for lower air temperatures, reflecting practical HVAC operation where cooler supply air decreases compressor demand [14].

3.3.2 Season-adaptive energy calculation

The thermal energy term T_i^s adapts to seasonal HVAC modes, accounting for heating versus cooling:

Summer season (June–August): In summer, thermal energy is typically negative, representing cooling load. The absolute value is used to quantify cooling energy, as heating is negligible in temperate climates [60]. The objective function prioritizes electrical consumption (fans, compressors):

$$\begin{aligned} T_i^{\text{summer}} &= |\text{thermal}_i| \quad \text{(cooling energy)} \\ f_{\text{summer}}(x) &= 0.7 \sum_{i=1}^{H} E_i + 0.3 \sum_{i=1}^{H} T_i^{\text{summer}} + 100 \sum_{i=1}^{H} \phi(PMV_i) + 5S(x) \\ &\quad - 0.1 \sum_{i=1}^{H} \max(30 - t_i, 0) \end{aligned} \tag{7}$$

Winter season (December–February): Heating dominates, with positive thermal energy values. The objective emphasizes thermal energy:

$$\begin{aligned} T_i^{\text{winter}} &= \max(\text{thermal}_i, 0) \quad \text{(heating energy)} \\ f_{\text{winter}}(x) &= 0.4 \sum_{i=1}^{H} E_i + 0.6 \sum_{i=1}^{H} T_i^{\text{winter}} + 100 \sum_{i=1}^{H} \phi(PMV_i) + 5S(x) \end{aligned} \tag{8}$$

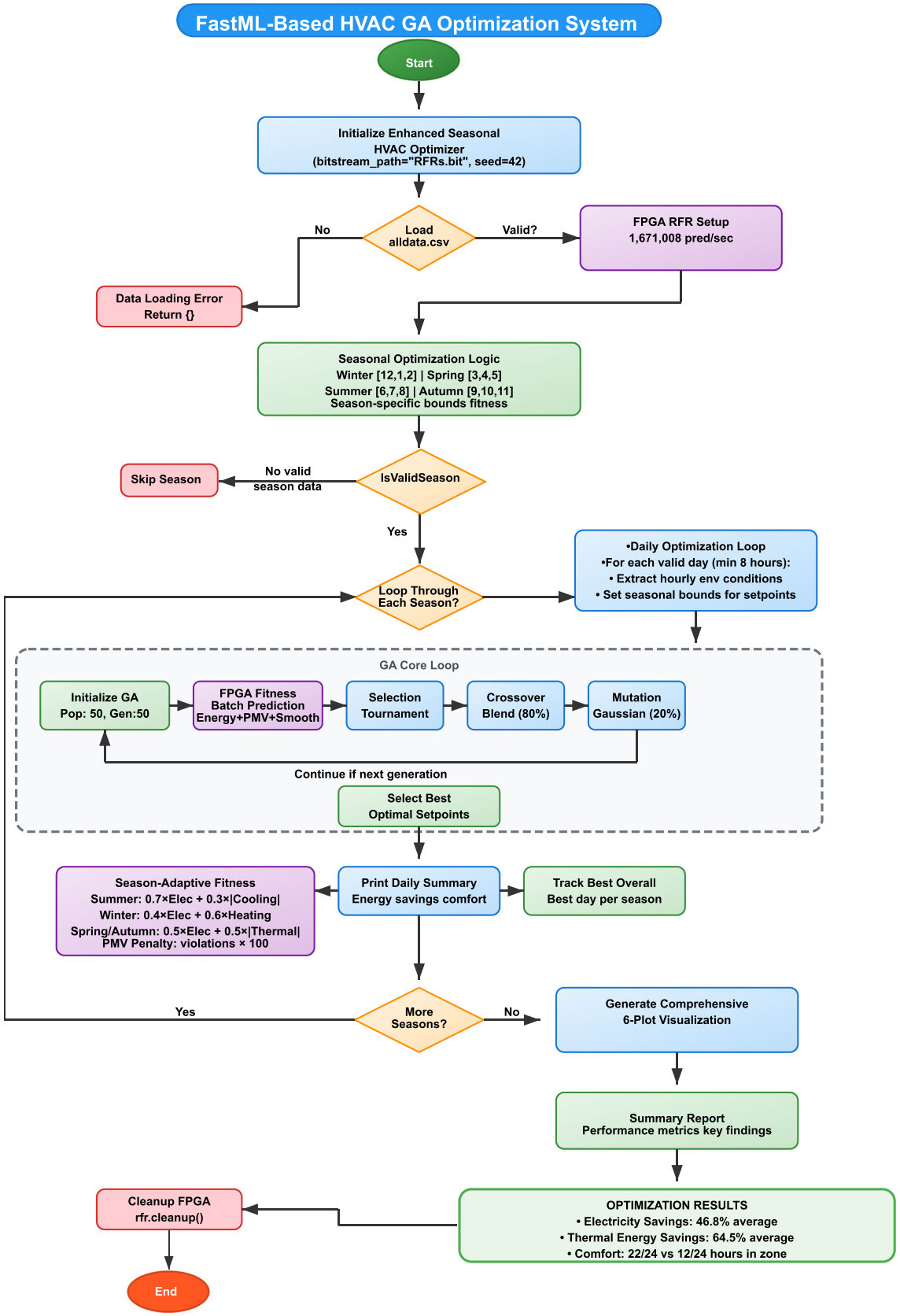

Fig. 3 Flowchart of the FastML-Based HVAC GA Optimization System, showing the hybrid PS–PL deployment: the GA executes on the PS (ARM Cortex-A9), while the RF surrogate model runs on the FPGA (PL) for batched fitness evaluations

Spring/Autumn seasons (March–May, September–November): A balanced approach handles mixed heating/cooling:

$$\begin{aligned} T_i^{\text{spring/autumn}} &= |\text{thermal}_i| \quad \text{(mixed modes)} \\ f_{\text{spring/autumn}}(x) &= 0.5\sum_{i=1}^{H} E_i + 0.5\sum_{i=1}^{H} T_i^{\text{spring/autumn}} + 100\sum_{i=1}^{H}\phi(PMV_i) + 5S(x) \end{aligned} \tag{9}$$

3.3.3 Season-specific operational constraints

Operational bounds, informed by ASHRAE Standard 55-2020 [14, 15], adapt to seasonal climatic conditions (Table 6):

For unoccupied periods, bounds are tightened to reduce energy use while maintaining minimal ventilation:

$$f_{\text{unoccupied}} \in [f_{\min,s} - 0.3, f_{\min,s}] \tag{10}$$

$$t_{\text{unoccupied}} \in [t_{\min,s} - 2, t_{\min,s}] \tag{11}$$

These constraints ensure compliance with thermal comfort standards and energy efficiency goals, adapting to occupancy patterns detected in the input dataset.

3.3.4 Enhanced comfort and smoothness penalties

Thermal comfort is enforced using the Predicted Mean Vote (PMV) model, with a comfort range of $[-0.5, 0.5]$ per ASHRAE Standard 55-2020 [14], corresponding to less than 10% predicted percentage of dissatisfied occupants. The comfort penalty counts violations:

$$\phi(PMV_i) = 100 \cdot 1_{\{|PMV_i|>0.5\}} \tag{12}$$

The high weight (100) prioritizes occupant satisfaction, penalizing any hour outside the comfort zone to reflect practical building management priorities [61]. The binary penalty ensures robust comfort enforcement over marginal PMV improvements within the acceptable range.

The smoothness penalty promotes stable HVAC operation, reducing wear on equipment:

$$S(x) = 5\left[\sum_{i=2}^{H}|f_i - f_{i-1}| + \sum_{i=2}^{H}|t_i - t_{i-1}|\right] \tag{13}$$

This penalty, weighted at 5, balances energy savings with operational stability, preventing abrupt setpoint changes that could stress HVAC components.

Table 6 Season-specific HVAC operational bounds

Season	Months	Air flow rate (m^3/s)	Air temperature (°C)	Energy focus
Winter	Dec, Jan, Feb	[1.2, 2.7]	[20, 25]	Heating (60%)
Spring	Mar, Apr, May	[1.3, 2.7]	[18, 24]	Balanced (50/50)
Summer	Jun, Jul, Aug	[1.5, 2.7]	[16, 24]	Cooling (70%)
Autumn	Sep, Oct, Nov	[1.4, 2.5]	[18, 26]	Balanced (50/50)

Table 7 Genetic algorithm parameters

Parameter	Value
Population size	50 individuals
Number of generations	50
Elite ratio	30% (15 individuals)
Crossover rate	0.8 (uniform crossover)
Mutation rate	0.2
Mutation distribution	Gaussian, $\sigma = 0.1$
PMV penalty weight	100
Smoothness weight	5
Working hours	08:00–17:00 (10 h)
Decision variables	20 (10 h $\times$ 2 setpoints)

3.3.5 Enhanced genetic algorithm parameters

The GA uses a robust configuration to ensure convergence to high-quality solutions, with parameters tuned for the HVAC optimization problem (Table 7):

The population size (50) and generations (50) are larger than typical GA settings (e.g., 20–30 individuals, 20–30 generations [62]), enhancing exploration of the 20-dimensional search space. The elite ratio (30%) preserves top solutions, while crossover (0.8) and mutation (0.2) balance exploitation and exploration. The Gaussian mutation with $\sigma = 0.1$ introduces controlled perturbations, ensuring fine-tuned adjustments within seasonal bounds.

Algorithm 2 Seasonal FPGA-accelerated genetic algorithm for HVAC optimization based on PS–PL

Require: Seasonal data `D_s`, environmental conditions `X_env`, season `s`, FPGA model `M`.
Ensure: Optimized seasonal setpoints x_s^*.
Load seasonal data for months corresponding to season `s`
Extract valid days with minimum 8 hours of data
Initialize seasonal bounds per Table 7
for each valid day `(month, day)` **do**
    Extract hourly environmental conditions `X_env` for hours 8–17
    Set adaptive bounds based on occupancy
    Initialize population `P` $= \{x_1, x_2, \ldots, x_{50}\}$ within bounds
    for `g` $= 1$ to 50 **do**
        Normalize `P` using min-max ranges (e.g., occupancy [0, 11], humidity [21.5, 100])
        Perform FPGA batch prediction `Y = M(P)`
        Compute fitness `f_s(x)` using Eq. (6)
        Select top 15 elite individuals
        Generate 35 offspring via crossover and mutation
        Form new population from elites and offspring
    end for
    Store daily optimization results
end for
Aggregate seasonal performance metrics
return best performing daily setpoints x_s^*

4 Results and discussion

The FPGA-accelerated genetic algorithm optimization was applied across all four seasons using 18 valid days per season from the HVAC dataset. The system demonstrated consistent performance improvements across different seasonal conditions, with electricity savings ranging from 49.6% to 56.4% and significant thermal energy reductions in heating and cooling dominated periods.

4.1 Validation of hardware prediction accuracy

Figure 4 compares the mean predicted values from the Python-based random forest model (scikit-learn) and our hardware-accelerated FastRFR framework deployed on FPGA. The near-perfect alignment of both curves across 200 test samples demonstrates excellent consistency between CPU and FPGA predictions, validating the correctness of the model export process and fixed-point quantization. The "mean predicted value" represents the average of the model's three outputs: electricity consumption, thermal energy use, and PMV, for easier interpretability.

In addition to high accuracy, our FastRFR framework achieved a peak inference throughput of 1,671,008 predictions/sec on the PYNQ-Z1 platform using `ap_fixed<16,6>`, with an on-chip power consumption of 1.821 W and an energy efficiency of 1.089 μJ/sample. This significantly outperforms prior work [57], which reported 490,196 predictions/sec, 1.824 W power, and 3.721 μJ/sample on the same hardware. FastRFR's superior performance stems from aggressive HLS optimizations, including initiation interval (II) = 1 pipelining, full array partitioning, stream-based dataflow, and DSP-accelerated fixed-point arithmetic. These choices minimize latency and resource overhead—unlike the DMA-controlled, 5-stage Decision Tree Processor (DTP) architecture used in that study.

The performance gap can be attributed to architectural differences. The DTP design incorporates a finite state machine (FSM) in the DECODE stage and shared memory controllers (e.g., BRAM in the MEM ACCESS stage), which introduce control complexity and inference latency. FSMs inherently incur overhead due to clock-driven state transitions, increasing power consumption and delay [63, 64]. Moreover, RAM-based FSM implementations often suffer from performance limitations due to memory decoding and access latency [65]. While the DTP pipeline reports a two-cycle BRAM access latency, per-decision latency is not specified. Nonetheless, the use

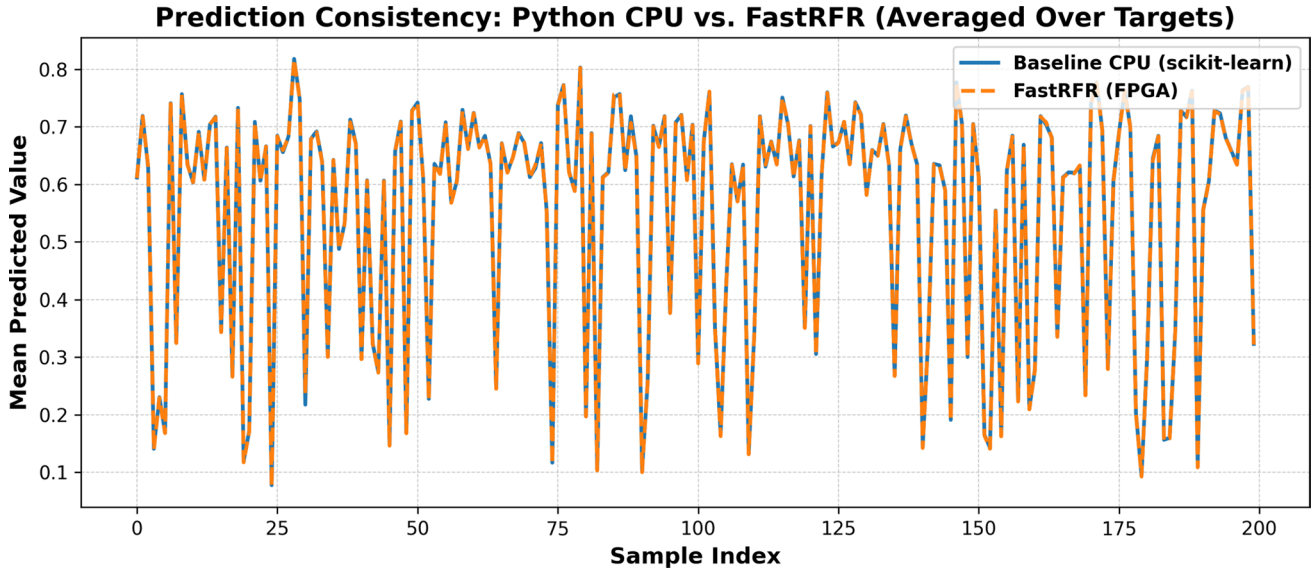

Fig. 4 Comparison of mean predicted values from the Python-based random forest (scikit-learn) and the FPGA-deployed FastRFR model. The close alignment across 200 test samples confirms high prediction consistency and accurate model export. Mean values are averaged over electricity, thermal energy, and PMV outputs

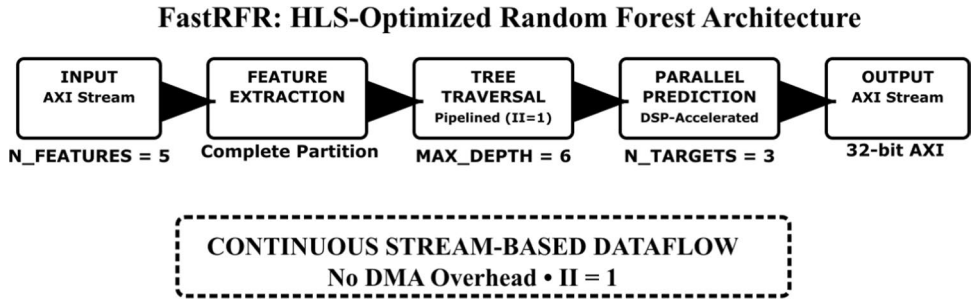

Fig. 5 Architecture of the FastRFR framework for random forest acceleration. The design integrates several HLS optimization directives: *Pipelined (II=1)* = loop pipelining with initiation interval of one cycle; *Array Partitioning* = dividing arrays into smaller memories for parallel access; *Cyclic Partitioning* = distributing array elements across memory banks cyclically; *DSP-Accelerated Arithmetic* = mapping arithmetic operations to dedicated DSP slices

Table 8 Sensitivity analysis of GA parameters on electricity savings and comfort hours

Pop. size	Mutation rate	PMV weight	Savings (%)	Comfort hours
10	0.05	50	41.3	7
20	0.05	50	53.2	9
40	0.05	50	47.3	8
10	0.10	100	47.3	8
20	0.10	100	53.2	9
40	0.10	100	41.3	7
10	0.20	200	53.2	9
20	0.20	200	47.2	8
40	0.20	200	53.2	9

of FSM-based stage control and memory arbitration adds architectural overhead—particularly in streaming and parallel processing scenarios.

These results underscore not only the functional correctness of the exported FastRFR model, but also its architectural efficiency achieving higher throughput and energy savings through a lean, stateless, and dataflow-optimized FPGA implementation (see Fig. 5).

4.2 Sensitivity analysis

To evaluate the robustness of the proposed optimization framework, we conducted a sensitivity analysis by varying key GA parameters, including population size, mutation rate, and the PMV penalty weight in the objective function. Table 8 reports the detailed results, and Fig. 6 provides a heatmap visualization. The results show that the framework consistently achieves electricity savings in the range of 41–53% across different GA configurations, while maintaining 7–9 comfort hours within the accepted thermal comfort range. Savings remain close to 50% even under extreme parameter variations, confirming that the optimization performance is robust and not overly sensitive to GA parameter tuning.

In terms of scalability, although experiments were conducted on the PYNQ-Z1 board, which is among the smallest FPGA platforms with limited resources, the design fit comfortably and consumed very low resources due to the optimizations applied in the random forest accelerator. On higher-capacity industrial FPGAs (e.g., Xilinx Virtex UltraScale or Intel Stratix families), the framework can scale efficiently by instantiating multiple parallel inference engines and supporting larger GA populations, thereby improving throughput and solution quality. To further validate adaptability, simulations were conducted under high-load and edge-case conditions, including occupancy spikes, heat waves, and cold snaps across different seasons. Results demonstrated that the framework consistently maintained stable optimization, ensuring PMV remained within the accepted thermal comfort band

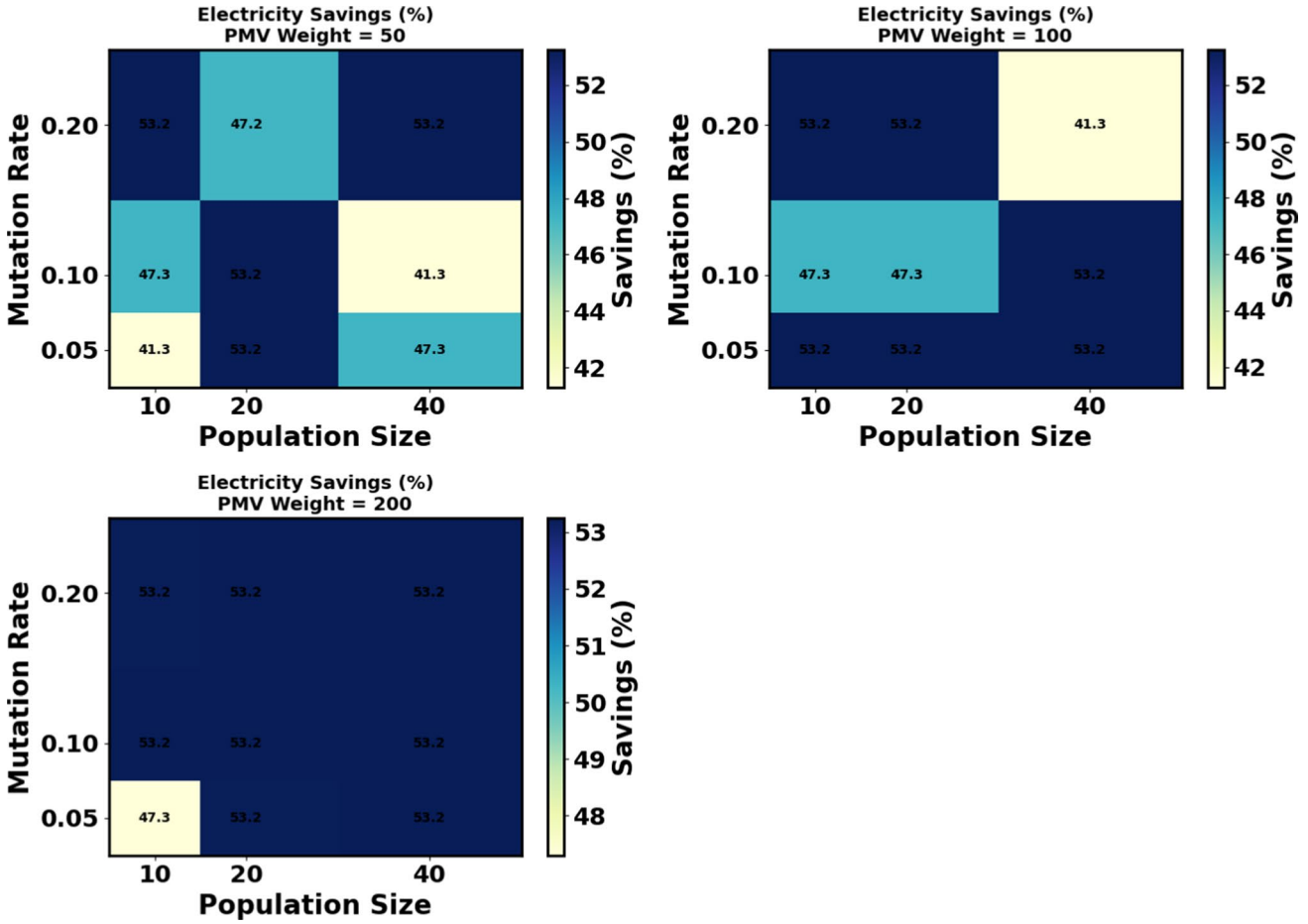

Fig. 6 Sensitivity analysis of GA parameters (population size, mutation rate, PMV weight) on electricity savings. Results show consistently high savings across configurations, confirming the robustness of the optimization framework

and achieving significant reductions in both electricity and thermal energy consumption compared to baseline operation. These findings confirm the resilience of the proposed system under demanding operating conditions.

Although the experiments in this work were conducted using a simulation-derived dataset, the proposed framework is readily adaptable to real-time deployment. In practice, live sensor streams (e.g., temperature, humidity, occupancy, CO_2, and energy metering) can be fed directly into the ARM processing system (PS) of the PYNQ-Z1 or a higher-capacity FPGA platform. These signals may be interfaced via standard IoT protocols or through a middleware layer such as a Raspberry Pi, as demonstrated by Mshragi et al. [29] for real-time smart meter data preprocessing and FPGA-based energy prediction in building management systems. The FPGA-accelerated random forest then processes incoming data in real time to predict energy consumption and comfort indices, while the genetic algorithm running on the PS evaluates and selects optimal HVAC setpoints. The resulting setpoints are transmitted to the building controllers for actuation, thereby completing the loop from acquisition to prediction, optimization, and control.

In our experiments, the optimization was performed with an hourly granularity, which aligns with practical building control horizons and minimizes unnecessary switching. However, due to the high inference throughput of the FPGA implementation (>1.6M predictions/s), the framework can also support finer resolutions (e.g., 15 min) without computational bottlenecks. This flexibility enables the system to respond effectively to rapid variations in occupancy or environmental conditions. The proposed framework incorporates several mechanisms for real-time robustness. First, operating on hourly windows (08:00–18:00) inherently smooths high-frequency

fluctuations and noise. Second, error handling and data validation are integrated into the pipeline: missing or corrupted records are skipped, and safe fallback values are returned in case of FPGA prediction errors. These mechanisms allow the optimizer to continue functioning under imperfect data conditions. For deployment with live sensors, the same data interface used for simulation can be connected to real-time measurements, with additional filtering and anomaly detection modules incorporated if required. Finally, because the framework is modular—separating prediction (random forest acceleration), optimization (genetic algorithm), and control into independent components with standardized input–output interfaces—it can be integrated with commercial BMS platforms. In this way, it can be deployed as an edge-level optimization module that operates alongside existing building control systems rather than replacing them, thereby facilitating practical integration into legacy infrastructure.

Although our framework was demonstrated on a single-zone HVAC system, it is readily extensible to multi-zone scenarios due to its modular design and low FPGA resource utilization (32.1% BRAM, 2.7% DSP, 6.3% LUT). The random forest accelerator, implemented as an overlay on the PYNQ-Z1 development board (which integrates a Xilinx Zynq-7020 SoC), currently processes five input features (occupancy, relative humidity, room temperature, air flow rate, and supply air temperature) using AXI-Stream DMA and HLS-optimized parallel inference in fixed-point arithmetic (`ap_fixed<16,6>`). For a four-zone configuration, this input expands to 20 features, which can be handled by replicating inference cores or batching inputs without requiring a redesign of the architecture. Given the Zynq-7020's available resources ($\approx 53k$ LUTs, 220 DSPs), the estimated utilization remains well within capacity when scaled to four zones, particularly with quantization optimizations, thereby preserving sub-millisecond inference latency. The GA, which currently optimizes 20 decision variables per zone (temperature and airflow setpoints), naturally scales to higher dimensions (e.g., 80 variables for four zones) by increasing the population size, as validated in our sensitivity analysis. Inter-zone dependencies can be incorporated into the fitness function through thermal coupling terms, ensuring balanced optimization of comfort and energy across zones. Finally, hardware extensions such as motorized dampers for zone airflow control and wireless sensors for real-time data collection can be seamlessly integrated via the existing GPIO, UART, or Ethernet interfaces. These characteristics confirm that the proposed framework can be adapted to multi-zone HVAC systems in large buildings, further enhancing its generality and practical applicability. Moreover, future extensions could incorporate additional contextual factors that strongly influence HVAC performance and occupant comfort. For example, solar radiation data would enable the optimizer to anticipate passive heat gains and adjust cooling loads accordingly. Dynamic energy pricing signals could be integrated to shift demand to lower-cost periods without sacrificing comfort. Similarly, building usage patterns (e.g., occupancy schedules or equipment usage) provide valuable prior knowledge for constraining optimization decisions and reducing unnecessary setpoint changes. In large public or commercial buildings, socio-cultural comfort factors (such as regional adaptive comfort expectations, clothing habits, or varying comfort ranges across different occupant groups) could be incorporated into the fitness function to better align energy savings with human well-being.

4.3 Seasonal performance overview

Figure 7 presents a comprehensive four-panel analysis of the seasonal HVAC optimization performance. The top-left panel demonstrates remarkably consistent electricity consumption reductions across all seasons, with winter achieving 51.2% savings, spring 50.0%, summer 49.6%, and autumn 51.7%. This consistency validates the robustness of the genetic algorithm approach regardless of seasonal variations in environmental conditions and operational requirements.

The top-right panel illustrates thermal energy reduction patterns, where seasonal differences become pronounced due to varying heating and cooling demands. Winter demonstrates the highest absolute thermal energy savings at 150.3 kWh, reflecting the substantial heating loads characteristic of cold weather operations. Spring follows with 100.9 kWh savings, indicating moderate thermal requirements during transitional weather. Autumn achieves 74.8 kWh savings as buildings transition from cooling to heating modes. Notably, summer shows "N/A"

 https://doi.org/10.1007/s00521-025-11737-x

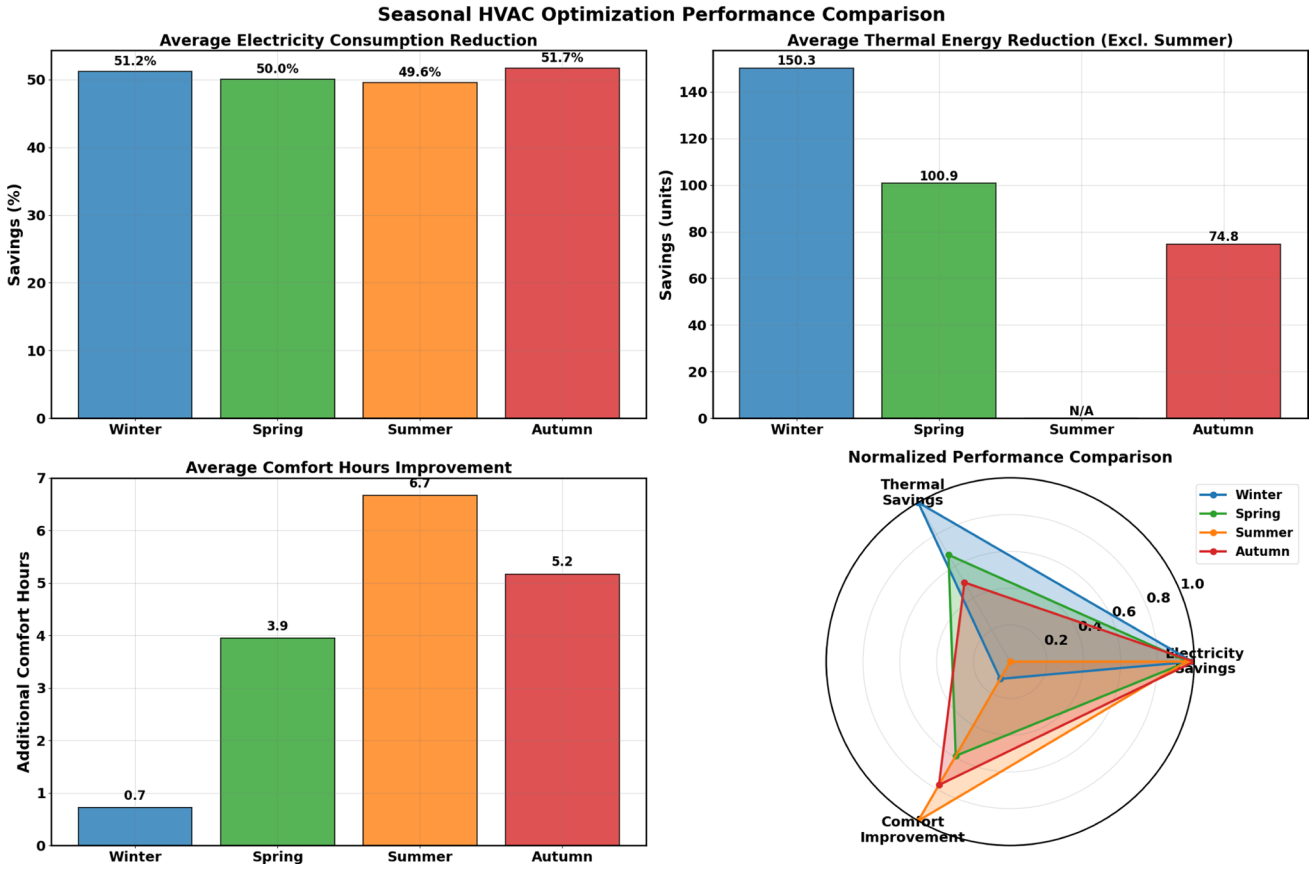

Fig. 7 Seasonal HVAC optimization performance comparison: (Top-left) Electricity consumption reduction showing consistent 49.6–51.7% savings across seasons; (Top-right) Thermal energy reduction with highest savings in winter (150.3 kWh) and summer marked N/A due to minimal thermal energy use; (Bottom-left) Comfort hour improvements ranging from 0.7 (winter) to 6.7 (summer); (Bottom-right) Normalized radar chart comparing multi-dimensional performance with autumn and spring showing most balanced profiles

for thermal savings because the dataset analysis revealed negligible thermal energy consumption during peak summer periods, where HVAC systems operate almost exclusively in cooling mode with minimal thermal energy requirements.

The bottom-left panel reveals significant variations in comfort hour improvements across seasons. Summer optimization achieved the most dramatic enhancement with 6.7 additional comfort hours, demonstrating the algorithm's effectiveness in managing challenging high-temperature conditions where baseline systems frequently failed to maintain acceptable PMV levels. Autumn follows with 5.2 additional comfort hours, spring with 3.9 h, and winter with 0.7 h. The modest winter improvement reflects the inherently better baseline comfort performance during heating-dominated periods, where thermal management is generally more predictable than cooling operations.

The bottom-right radar chart provides a normalized multi-dimensional performance comparison across all optimization objectives. Each axis represents a different performance metric scaled from 0 to 1, where values closer to the perimeter indicate superior performance. The electricity savings axis shows all seasons performing excellently with values near 1.0, confirming consistent energy efficiency gains. The thermal savings axis clearly distinguishes heating-dominated seasons (winter and autumn) from cooling-dominated periods (summer showing minimal values). The comfort improvement axis highlights summer's exceptional performance in addressing thermal discomfort challenges. The radar visualization reveals that autumn and spring achieve the most balanced

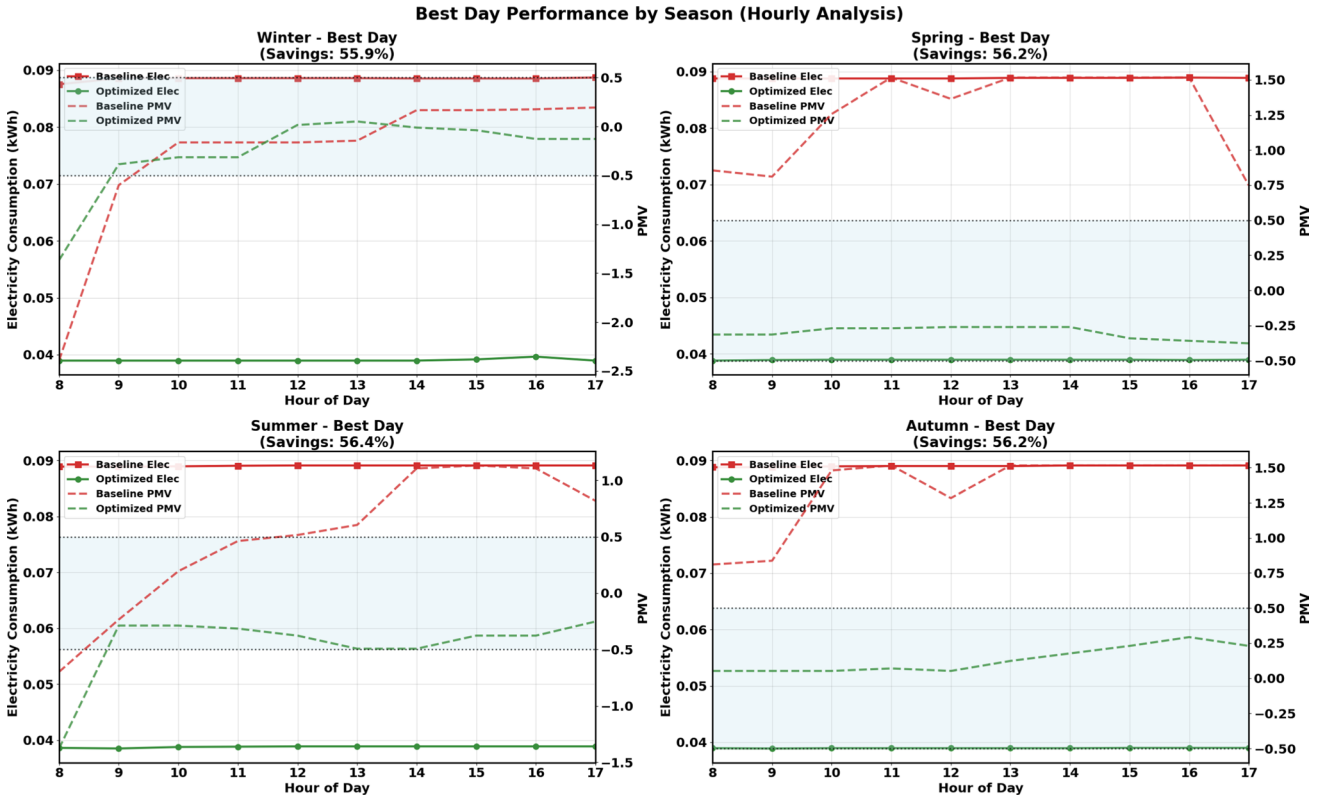

Fig. 8 Best day performance by season (hourly analysis): Winter (M01D05), Spring (M05D03), Summer (M06D06), and Autumn (M09D04) showing electricity consumption, PMV comfort levels, and optimization effectiveness throughout 08:00–17:00 operational periods. Each plot demonstrates season-specific optimization characteristics with consistent energy savings and improved thermal comfort management

performance profiles, excelling across multiple dimensions simultaneously, while winter and summer show more specialized optimization characteristics aligned with their dominant thermal loads.

4.4 Best day hourly performance analysis

Figure 8 presents the hourly performance dynamics for the best performing day in each season, illustrating how optimization strategies adapt to different temporal and seasonal patterns. Each seasonal subplot displays electricity consumption (solid lines), PMV comfort levels (dashed lines), and their corresponding optimization trajectories throughout the 10-h operational period from 08:00 to 17:00.

Winter's best day (M01D05, 55.9% savings) demonstrates dramatic morning energy reductions where electricity consumption drops from 0.088 kWh to 0.039 kWh during peak heating startup periods (08:00–10:00). The PMV profile shows successful comfort maintenance with optimized values consistently within the acceptable range (−0.5 to 0.5), while baseline conditions exhibit excessive cooling sensations below −0.5 during morning hours. The optimization achieves this through strategic reduction of air flow rates and temperature adjustments that minimize heat loss while preserving thermal comfort.

Spring's best day (M05D03, 56.2% savings) exhibits the most stable optimization performance with smooth energy transitions and exceptional comfort control. The baseline PMV values consistently exceed 0.5 (indicating warm discomfort), while optimization maintains perfect comfort throughout all operational hours. The electricity profile shows consistent reduction from approximately 0.089 kWh to 0.039 kWh, demonstrating the algorithm's ability to maintain steady performance during transitional weather conditions with mixed heating and cooling requirements.

Summer's best day (M06D06, 56.4% savings) focuses primarily on electrical consumption management due to minimal thermal energy requirements. The most significant improvements occur during peak thermal stress periods (12:00–16:00) where baseline PMV values exceed 1.0 (indicating hot discomfort), while optimization successfully maintains comfortable conditions around −0.25. The consistent electrical consumption reduction from 0.089 kWh to approximately 0.039 kWh throughout the day demonstrates effective cooling load management.

Autumn's best day (M09D04, 56.2% savings) shows balanced optimization across both energy dimensions with smooth PMV transitions. The algorithm effectively manages the transition from daytime cooling needs to evening heating requirements, maintaining comfort while achieving substantial energy savings. The PMV profile demonstrates sophisticated thermal control with optimized values remaining stable around −0.25 while baseline conditions show significant variation.

4.5 Individual seasonal trade-off analysis

The individual seasonal 3D Pareto analyses provide detailed visualization of the multi-objective optimization trade-offs achieved in each season, revealing distinct seasonal characteristics and optimization patterns.

4.5.1 Winter season analysis

Figure 9 illustrates winter optimization results with the best day (M01D05) achieving 55.9% electricity savings and substantial thermal energy reductions from 248.21 kWh to 125.91 kWh. The baseline points (red circles)

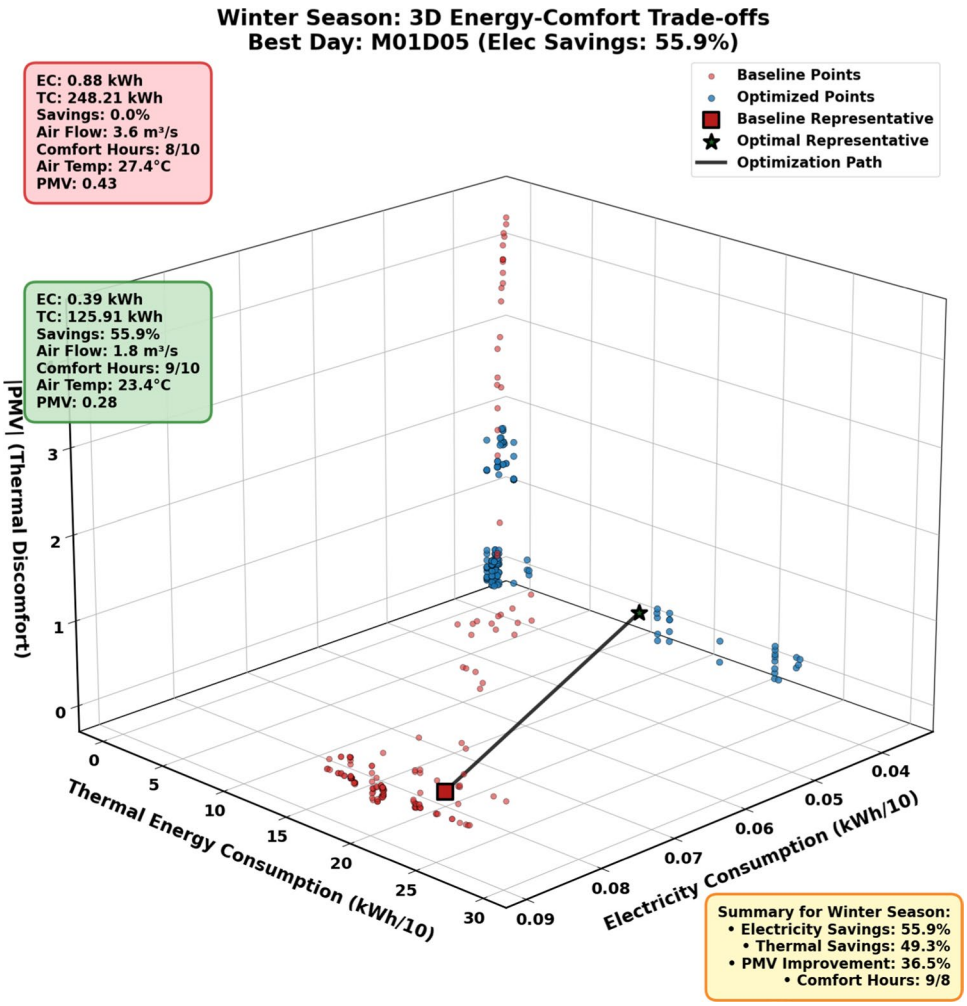

Fig. 9 Winter season: 3D energy-comfort trade-offs showing optimization from high thermal energy baseline (248.21 kWh) to efficient operation (125.91 kWh) with best day M01D05 achieving 55.9% electricity savings and maintaining 9/8 comfort hours

cluster in high thermal energy regions (15–30 kWh/10 h) with poor comfort performance, while optimized points (red circles) concentrate in the efficient zone with thermal consumption below 10 kWh/10 h. The clear optimization path from baseline representative (red square) to optimal representative (black star) demonstrates the algorithm's capability to navigate complex winter heating requirements while maintaining comfort constraints. The tight clustering of optimized points indicates consistent performance across different winter operating conditions.

4.5.2 Spring season analysis

Figure 10 demonstrates spring optimization characteristics with best day (M05D03) achieving 56.2% electricity savings and thermal reduction from 66.95 kWh to 1.99 kWh. The spring optimization shows excellent clustering of optimized points (green squares) in the low-energy, high-comfort region. The Pareto front analysis reveals superior convergence with optimized solutions forming a distinct cluster separated from baseline performance. The optimization path clearly indicates the algorithm's success in managing transitional weather requirements while achieving perfect comfort (10/10 h) compared to poor baseline performance (0/10 h).

4.5.3 Summer season analysis

Figure 11 presents unique summer optimization characteristics with best day (M06D06) achieving 56.4% electricity savings. The thermal energy dimension shows minimal values (near zero) reflecting cooling-dominated

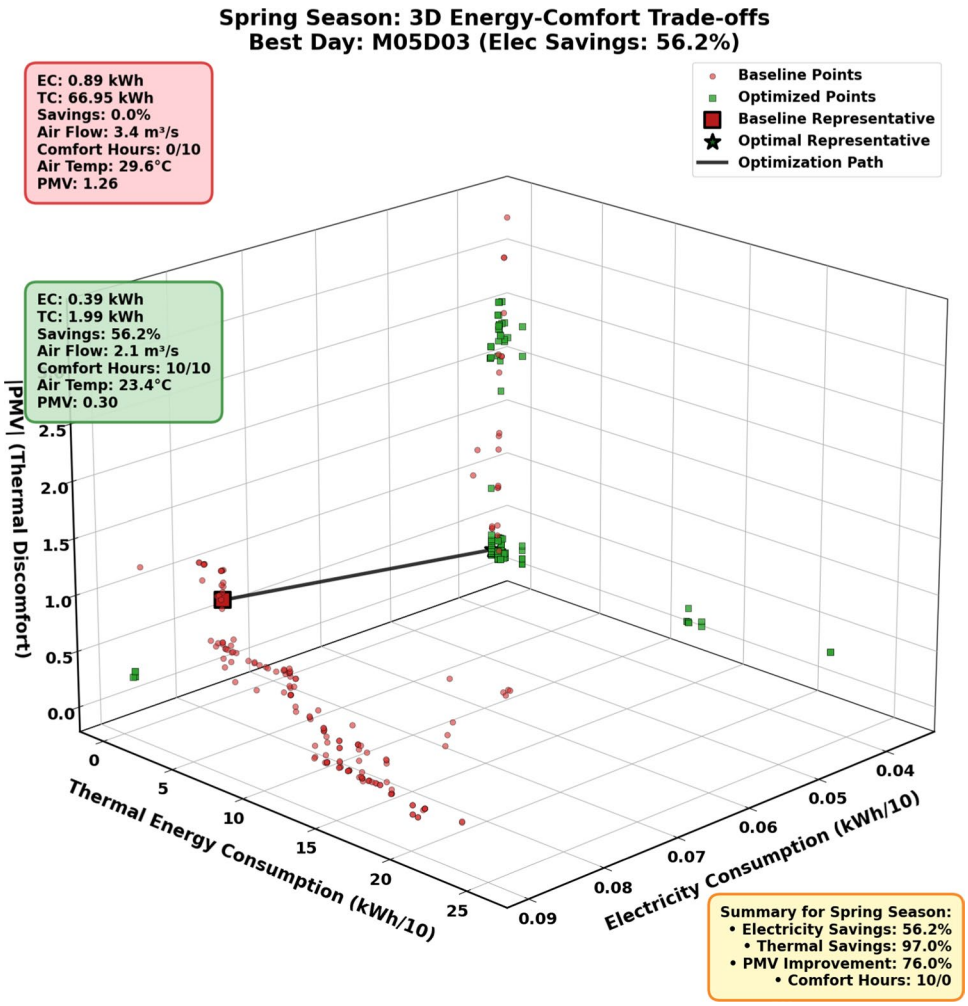

Fig. 10 Spring season: 3D energy-comfort trade-offs demonstrating excellent optimization clustering with best day M05D03 achieving 56.2% electricity savings, thermal reduction from 66.95 kWh to 1.99 kWh, and perfect comfort performance (10/10 h)

 https://doi.org/10.1007/s00521-025-11737-x

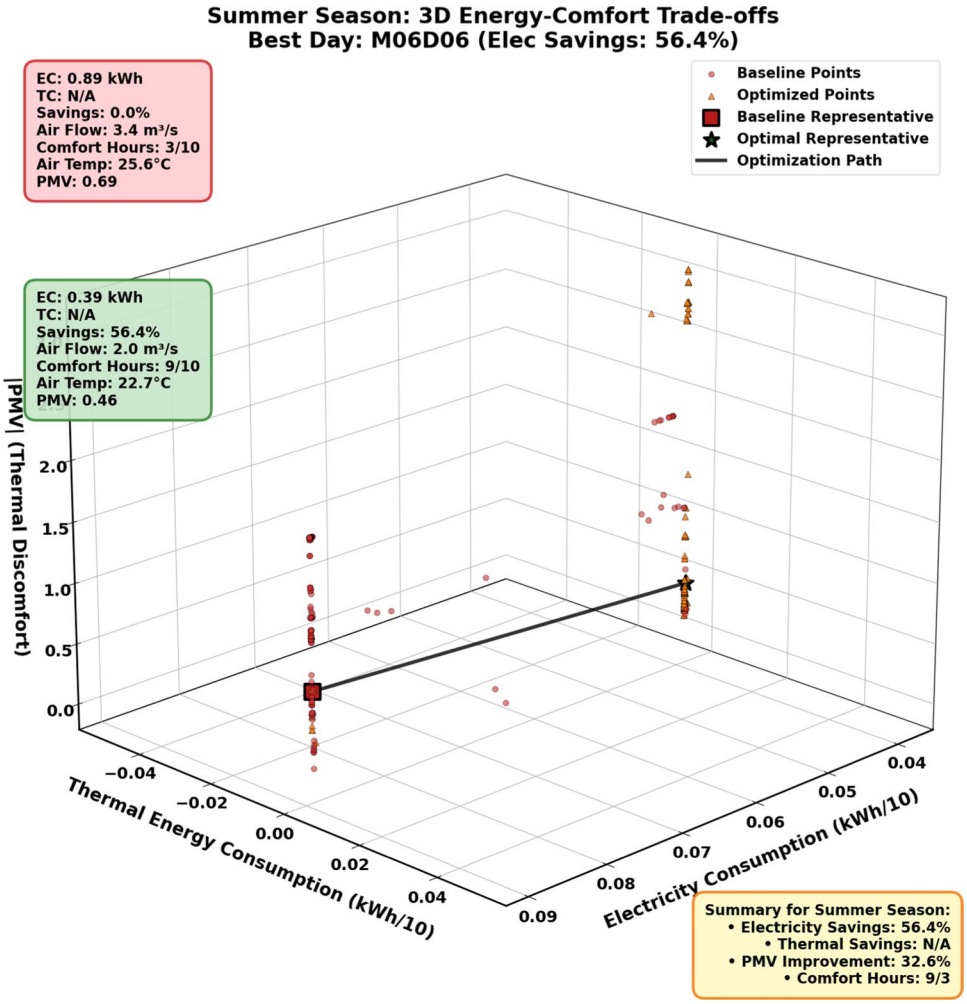

Fig. 11 Summer season: 3D energy-comfort trade-offs showing cooling-dominated optimization with minimal thermal energy (N/A baseline) and best day M06D06 achieving 56.4% electricity savings with dramatic comfort improvement from 0% to 56.4% comfort hours

operations where thermal energy consumption is negligible. Optimized points (orange triangles) cluster in the minimal thermal energy region while achieving significant comfort improvements from 0.0% to 56.4% comfort hours. The baseline thermal consumption shows "N/A" indicating virtually no thermal energy requirements during summer cooling operations, validating the seasonal optimization strategy focus on electrical consumption.

4.5.4 Autumn season analysis

Figure 12 reveals autumn optimization results with best day (M09D04) achieving 56.2% electricity savings and thermal reduction from 31.23 kWh to 1.91 kWh. The autumn optimization demonstrates balanced performance across energy dimensions with optimized points (red diamonds) achieving superior clustering in the efficient zone. The 93.9% thermal savings and 90.1% PMV improvement indicate excellent optimization effectiveness during transitional autumn conditions. The clear separation between baseline and optimized regions confirms the algorithm's adaptability to mixed heating/cooling requirements.

4.6 Multi-seasonal comparative analysis

Figure 13 provides a comprehensive view of all seasonal optimization results in a single 3D space, enabling direct comparison of seasonal characteristics and optimization effectiveness. The multi-seasonal clustering reveals distinct patterns: winter solutions (red circles) occupy high thermal energy regions (20–25 kWh/10 h) reflecting

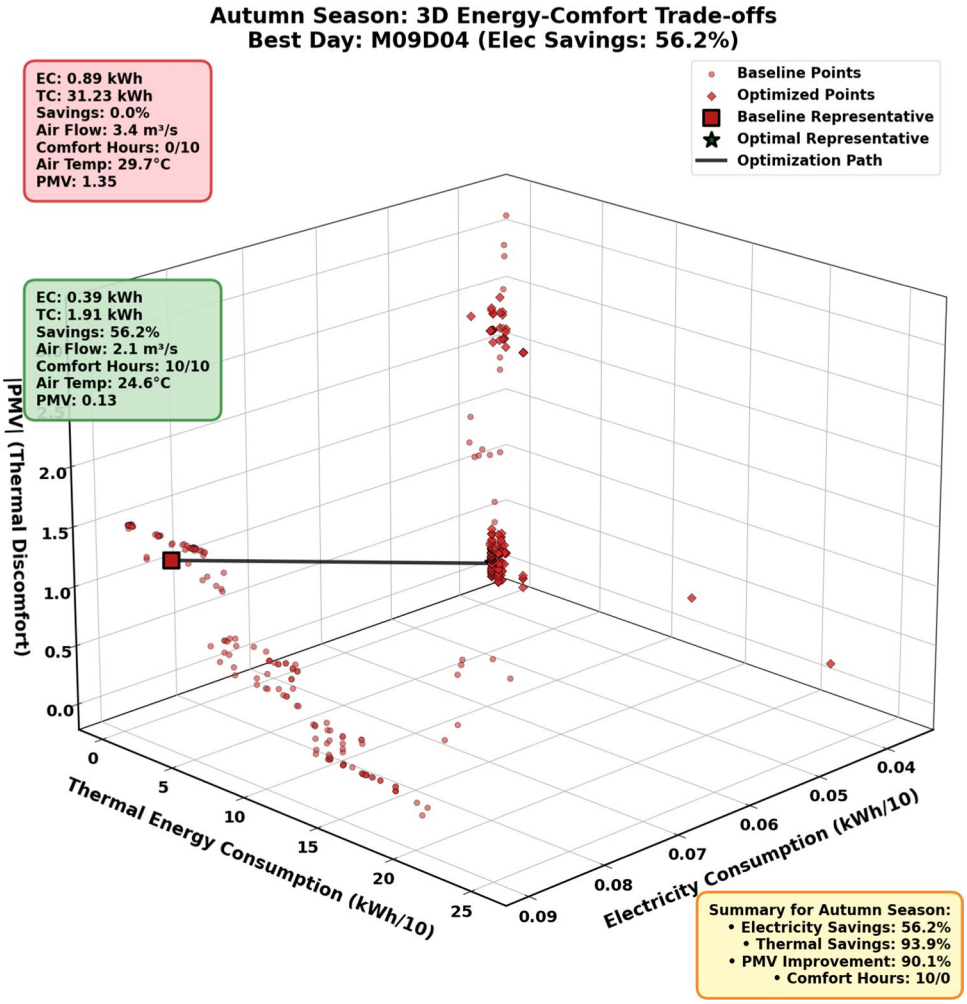

Fig. 12 Autumn season: 3D energy-comfort trade-offs showing balanced optimization with best day M09D04 achieving 56.2% electricity savings, 93.9% thermal savings, and 90.1% PMV improvement during transitional weather conditions

heating requirements, summer solutions (orange triangles) cluster near zero thermal energy confirming cooling-dominated operations, while spring (green squares) and autumn (red diamonds) show intermediate thermal energy levels consistent with mixed heating/cooling demands.

The comparative analysis demonstrates the genetic algorithm's remarkable adaptability to diverse seasonal conditions while maintaining consistent electricity savings performance across all seasons. Each season forms distinct clusters in the 3D space, validating the seasonal optimization strategy and confirming that the algorithm successfully identifies season-appropriate solutions. The comfort dimension improvements are evident across all seasons, with each achieving substantial PMV enhancements compared to baseline operations.

This comprehensive seasonal analysis validates the effectiveness of the FPGA-accelerated genetic algorithm approach for HVAC optimization across diverse operating conditions. The system demonstrates both consistency in electricity savings (49.6−51.7%) and remarkable adaptability to seasonal requirements. Thermal energy optimization shows appropriate seasonal variation, with substantial savings in heating-dominated periods and minimal consumption during cooling-dominated summer operations. Comfort improvements are significant across all seasons, with particularly impressive gains during challenging summer conditions. The 3D Pareto analyses confirm the algorithm's ability to navigate complex multi-objective trade-offs while maintaining operational constraints and achieving superior performance compared to baseline operations.

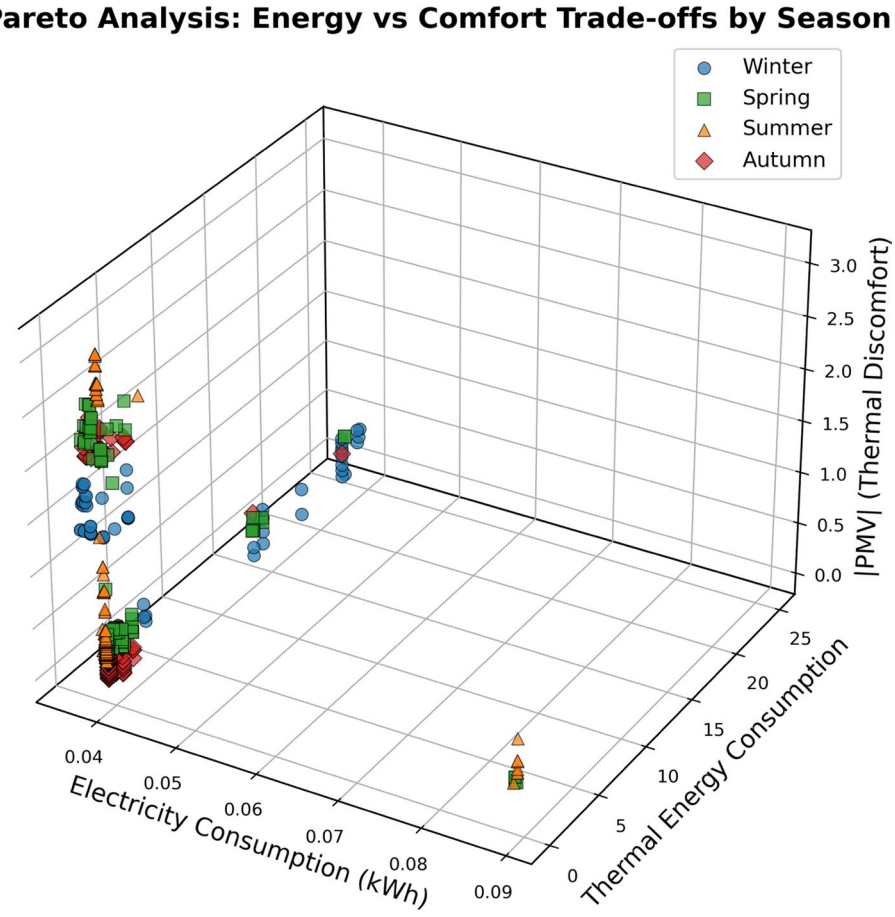

Fig. 13 Multi-seasonal 3D Pareto analysis: Comprehensive comparison showing distinct seasonal clustering with winter (red) in high thermal zones, summer (orange) near zero thermal energy, and spring/autumn (green/red) in intermediate regions, demonstrating algorithm adaptability across diverse operating conditions

5 Conclusion

This paper introduced FastML-GA, a novel FPGA-accelerated framework designed to optimize energy and HVAC systems by integrating embedded machine learning with a genetic algorithm. Experimental evaluations across multiple seasons demonstrated significant and consistent electricity savings, with an average reduction of 46.8% and up to 56% on the best-performing days, contributing to building decarbonization trajectories without compromising comfort, highlighting the framework's practical effectiveness.

The novelty of FastML-GA lies in its hybrid PS–PL architecture, where the random forest surrogate model is implemented as a hardware accelerator on the FPGA's programmable logic (PL), while the lightweight genetic algorithm executes on the embedded ARM Cortex-A9 processors of the processing system (PS). This integration achieves inference throughput rates surpassing 1.67 million predictions per second and enables real-time, fine-grained optimization at the edge, effectively addressing the latency and computational efficiency constraints typical of embedded building management systems.

The importance of this research extends beyond technical novelty, offering a substantial practical contribution to the field of intelligent building management. By significantly improving energy efficiency and occupant comfort simultaneously, the framework addresses key sustainability and operational challenges facing contemporary buildings and aligns with decarbonization targets through reduced operational energy use. Consequently, FastML-GA represents a meaningful advancement towards more responsive, adaptive, and energy-conscious building environments that balance occupant well-being with environmental conditions.

The FastML-GA framework demonstrates significant potential for enhancing HVAC efficiency in buildings through embedded machine learning and optimization. The FPGA-accelerated random forest regressor (RFR)

achieves sub-millisecond inference latency supporting up to 1.67 million predictions per second—enabling rapid adaptation to dynamic environmental and occupancy conditions.

The current evaluation relies on datasets that provide controlled benchmarking. However, real-time building deployments introduce challenges such as sensor noise, occupant behavior variability, and environmental uncertainties. To address sensor noise—which impacted `Therm_Eng_Cons` accuracy—we implemented hardware-based Min-Max normalization, allowing a reduction in fixed-point precision from `ap_fixed<18,12>` to `ap_fixed<16,6>` without sacrificing accuracy. This normalization mitigates sensor scale mismatches and target range disparities, enabling efficient inference that supports sustained energy savings in practice.

In practical situations, implementing aggressive lower setpoints during unoccupied times is difficult due to system delays and sensor uncertainties. However, FastML-GA facilitates quick adaptations to changes in occupancy and the environment, making energy-saving strategies reactive and more efficacious thus reducing avoidable energy use consistent with these targets. Furthermore, the ensemble nature of the random forest regression (RFR) boosts robustness to variability and noise, outperforming CPU-based reinforcement learning methods by enabling rapid, low-power, edge-based control that prioritizes both comfort and energy performance.

Dynamic building characteristics such as weather variation, equipment degradation, and maintenance events need to be considered to improve long-term adaptability. Evaluating deployment on diverse embedded platforms will help characterize trade-offs between energy efficiency, prediction speed, and system complexity. Expanding the surrogate model to include contextual features (e.g., occupancy schedules or user preferences) may further enhance predictive accuracy and system responsiveness in service of occupant comfort and carbon reduction.

5.1 Limitations and future work

Our hybrid FastML-GA FPGA implementation employs a fixed random forest regressor; however, model retraining can be performed offline and redeployed to the FPGA as updated parameters or bitstreams. This approach would allow the system to adapt over time to model drift caused by evolving occupancy patterns, equipment degradation, or changing environmental conditions while remaining consistent with decarbonization ambitions and occupant well-being. FastML-GA could be enhanced through integration with external predictive models, such as occupancy forecasting systems or weather prediction APIs, to further improve control robustness and energy savings without compromising comfort. In real deployments, sensors may drift, fail, or provide noisy readings; therefore, future extensions should incorporate redundancy (e.g., multiple sensors per key variable), anomaly detection, or fallback to safe default control strategies to ensure fault tolerance and uninterrupted comfort delivery. Since FastML-GA operates at the edge, sensitive data is processed locally, reducing exposure risks associated with transmission to centralized clouds. Nevertheless, cybersecurity and privacy remain critical, particularly in protecting FPGA bitstreams, control data, and sensitive occupancy information; secure data handling, authentication protocols, and encrypted FPGA configuration are therefore essential for resilient, trustworthy, and sustainable building operation.

In addition to these considerations, this study demonstrates the feasibility and performance benefits of FPGA-accelerated HVAC optimization under controlled simulation conditions. Future analyses should include validation using long-term, real-world building data with unpredictable occupancy patterns and sensor variability, which is essential to fully assess practical effectiveness and sustained impact on decarbonization pathways. Although the random forest predictor is inherently robust to moderate noise, lightweight preprocessing techniques (e.g., moving average smoothing or Kalman-based filtering) and systematic noise sensitivity tests can further reinforce reliability under real-world conditions. In practical deployments, additional resilience mechanisms are required to manage sensor drift, bias, and intermittent failures. Such approaches could include integrating moving average, exponential, or Kalman-based filters within the processing system (PS) pipeline prior to FPGA inference to smooth fluctuations, correct long-term drift, and reconstruct missing data. In addition, noise sensitivity experiments could be performed by injecting Gaussian noise, bias offsets, and dropout scenarios into the dataset, thereby quantifying how uncertainty propagates through the random forest surrogate and affects GA

optimization outcomes. Redundancy strategies (e.g., fusing multiple sensors for temperature or occupancy) and anomaly detection modules could further safeguard operation by detecting faulty measurements and reverting to safe fallback strategies. Together, these methods would ensure stable optimization decisions, preserving both occupant comfort and energy efficiency even under adverse sensing conditions. While the current design does not support online model adaptation, retraining can be performed offline and redeployed to the FPGA as updated parameters or bitstreams while incremental retraining strategies or dynamic partial reconfiguration may enable more flexible adaptation. The random forest surrogate could be extended with additional contextual features such as weather forecasts, occupancy schedules, or user preferences to enhance adaptability in situ. Finally, edge deployment provides advantages in latency reduction and local data processing, and creates opportunities to strengthen resilience through redundancy, fault detection, and secure data handling for supporting human-centric, energy-efficient operation consistent with these targets.

Acknowledgements Not applicable.

Author contributions Mohammed Mshragi developed the model and wrote the manuscript. Ioan Petri supervised the research, reviewed the manuscript, and provided critical revision and strategic guidance.

Funding No external funding was received for this study. Authors' contributions: Mohammed Mshragi developed the model and wrote the manuscript. Ioan Petri supervised the research, reviewed the manuscript, and provided critical revision and strategic guidance.

Data availability The datasets generated and/or analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare that they have no competing interests.

Ethical approval and consent to participate Not applicable.

Consent for publication Not applicable.

References

1. Hamilton I, Kennard H, Rapf O, Amorocho J, Steuwer S, Kockat J, Toth Z (2024) Global status report for buildings and construction—beyond foundations: mainstreaming sustainable solutions to cut emissions from the buildings sector. United Nations Environment Programme (UNEP) / Global Alliance for Buildings and Construction, Nairobi, Kenya. ISBN: 978-92-807-4131-5. https://globalabc.org/sites/default/files/2024-11/global_status_report_buildings_construction_2023.pdf
2. Wang H, Chen X, Vital N, Duffy E, Razi A (2024) Energy optimization for HVAC systems in multi-VAV open offices: a deep reinforcement learning approach. Appl Energy 356:122354
3. Hsu P-C, Gao L, Hwang Y, Radermacher R (2025) A review of the state-of-the-art data-driven modeling of building HVAC systems. Energy Build 115881
4. Zhou SL, Shah AA, Leung PK, Zhu X, Liao Q (2023) A comprehensive review of the applications of machine learning for HVAC. DeCarbon 2:100023. https://doi.org/10.1016/j.decarb.2023.100023

5. Kurte K, Munk J, Kotevska O, Amasyali K, Smith R, McKee E, Du Y, Cui B, Kuruganti T, Zandi H (2020) Evaluating the adaptability of reinforcement learning based HVAC control for residential houses. Sustainability 12(18):7727
6. Zhao D, Watari D, Ozawa Y, Taniguchi I, Suzuki T, Shimoda Y, Onoye T (2023) Data-driven online energy management framework for HVAC systems: an experimental study. Appl Energy 352:121921. https://doi.org/10.1016/j.apenergy.2023.121921
7. Ahmad MW, Mourshed M, Rezgui Y (2017) Trees vs neurons: comparison between random forest and ANN for high-resolution prediction of building energy consumption. Energy Build 147:77–89. https://doi.org/10.1016/j.enbuild.2017.04.038
8. Van Essen B, Macaraeg C, Gokhale M, Prenger R (2012) Accelerating a random forest classifier: multi-core, GP-GPU, or FPGA? In: 2012 IEEE 20th international symposium on field-programmable custom computing machines, pp 232–239. https://doi.org/10.1109/FCCM.2012.47
9. Aftab M, Chen C, Chau C-K, Rahwan T (2017) Automatic HVAC control with real-time occupancy recognition and simulation-guided model predictive control in low-cost embedded system. Preprint at arXiv:1708.05208
10. Choppali Sudarshan C, Arora A, Chhabria VA (2024) GREENFPGA: evaluating FPGAS as environmentally sustainable computing solutions. In: Proceedings of the 61st ACM/IEEE design automation conference, pp 1–6
11. Turley C, Jacoby M, Pavlak G, Henze G (2020) Development and evaluation of occupancy-aware HVAC control for residential building energy efficiency and occupant comfort. Energies 13(20):5396
12. Hozayen MI, Abass WH (2025) Optimized human-centric decision-maker for HVAC systems using genetic algorithms to establish an equilibrium point for enhanced thermal comfort. In: 15th international conference on environmental science and development (ICESD), pp 111–122. https://doi.org/10.1007/978-3-031-88683-6_9
13. Mshragi M, Petri I (2025) Fast machine learning for building management systems. Artif Intell Rev 58(7):211
14. ASHRAE: ANSI/ASHRAE Standard 55-2020: thermal environmental conditions for human occupancy. Technical report, American Society of Heating, Refrigerating and Air-Conditioning Engineers (2021). Pages: 80. https://www.ashrae.org/technical-resources/standards-and-guidelines/read-only-versions-of-ashrae-standards
15. International Organization for Standardization (1984) ISO 7730: moderate thermal environments-determination of the PMV and PPD indices and specification of the conditions for thermal comfort. Technical report, ISO, Geneva, Switzerland
16. Guo Y, Liu Y, Wang Z, Hu Y (2023) Application of data-driven methods for heating ventilation and air conditioning systems. MDPI 11(11):3133
17. Araújo GR, Gomes R, Gomes MG, Guedes MC, Ferrão P (2023) Surrogate models for efficient multi-objective optimization of building performance. Energies 16(10):4030
18. Ala'raj M, Radi M, Abbod MF, Majdalawieh M, Parodi M (2022) Data-driven based HVAC optimisation approaches: a systematic literature review. J Build Eng 46:103678
19. Wang Z, Wang Y, Zeng R, Srinivasan RS, Ahrentzen S (2018) Random forest based hourly building energy prediction. Energy Build 171:11–25
20. Ilbeigi M, Ghomeishi M, Dehghanbanadaki A (2020) Prediction and optimization of energy consumption in an office building using artificial neural network and a genetic algorithm. Sustain Cities Soc 61:102325
21. Ferreira P, Ruano A, Silva S, Conceicao E (2012) Neural networks based predictive control for thermal comfort and energy savings in public buildings. Energy build 55:238–251
22. Ozawa Y, Zhao D, Watari D, Taniguchi I, Suzuki T, Shimoda Y, Onoye T (2023) Data-driven HVAC control using symbolic regression: design and implementation. In: 2023 IEEE Power & Energy Society General Meeting (PESGM). IEEE, pp 1–5
23. Ding X, An Z, Rathee A, Du W (2025) A safe and data-efficient model-based reinforcement learning system for HVAC control. IEEE Internet Things J
24. Lu M, Sun Y, Ma Z (2024) Multi-objective design optimization of multiple energy systems in net/nearly zero energy buildings under uncertainty correlations. Appl Energy 370:123620
25. Xu C, Jiang S, Luo G, Sun G, An N, Huang G, Liu X (2022) The case for FPGA-based edge computing. IEEE Trans Mob Comput 21(7):2610–2619. https://doi.org/10.1109/TMC.2020.3041781
26. Vaithianathan M, Patil M, Ng SF, Udkar S (2024) Low-power FPGA design techniques for next-generation mobile devices. ESP Int J Adv Comput Technol(ESP-IJACT) 2(2):82–93
27. Qian C, Ling T, Cichiwskyj C, Schiele G (2025) Configuration-aware approaches for enhancing energy efficiency in FPGA-based deep learning accelerators. J Syst Architect 163:103410
28. Bartoli P, Veronesi C, Giudici A, Siorpaes D, Trojaniello D, Zappa F (2025) Benchmarking energy and latency in tinyml: a novel method for resource-constrained ai. Preprint at arXiv:2505.15622
29. Mshragi M, Petri I, Rana O (2025) FPGA-accelerated fast machine learning for heterogeneous edge systems. In: 2025 IEEE international conference on edge computing and communications (EDGE). IEEE, pp 183–191
30. Omidsajedi SN, Reddy R, Yi J, Herbst J, Lipps C, Schotten HD (2024) Latency optimized deep neural networks (DNNs): an artificial intelligence approach at the edge using multiprocessor system on chip (MPSOC). Preprint at arXiv:2407.18264

31. Qasaimeh M, Denolf K, Khodamoradi A, Blott M, Lo J, Halder L, Vissers K, Zambreno J, Jones PH (2021) Benchmarking vision kernels and neural network inference accelerators on embedded platforms. J Syst Architect 113:101896
32. Archet A, Gac N, Orieux F, Ventroux N (2023) Embedded ai performances of nvidia's jetson orin soc series. In: 17ème Colloque National du GDR SOC2
33. Fursin G (2024) Enabling more efficient and cost-effective AI/ML systems with collective mind, virtualized MLOps, MLPerf, and reproducible optimization tournaments. Technical report, cKnowledge.org and MLCommons. White Paper
34. Karumbunathan LS (2022) Nvidia jetson agx orin series. A Giant leap forward for robotics and edge AI applications. Technical Brief
35. Agouzoul A, Chegari B, Tabaa M, Simeu E (2022) Using neural network in a model-based predictive control loop to enhance energy performance of buildings. Energy Rep 8:1196–1207
36. Arun S, Selvan M (2019) Smart residential energy management system for demand response in buildings with energy storage devices. Front Energy 13(4):715–730
37. Almazam K, Humaidan O, Shannan NM, Bashir FM, Gammoudi T, Dodo YA (2025) Innovative energy efficiency in HVAC systems with an integrated machine learning and model predictive control technique: a prospective toward sustainable buildings. Sustainability 17(7):2916
38. Hanumaiah V, Genc S (2021) Distributed multi-agent deep reinforcement learning framework for whole-building HVAC control. Preprint at arXiv:2110.13450
39. Gao G, Li J, Wen Y (2019) Energy-efficient thermal comfort control in smart buildings via deep reinforcement learning. Preprint at arXiv:1901.04693
40. Li D, Qi Z, Zhou Y, Elchalakani M (2025) Machine learning applications in building energy systems: review and prospects. Buildings 15(4):648
41. Fang Z, Crimier N, Scanu L, Midelet A, Alyafi A, Delinchant B (2021) Multi-zone indoor temperature prediction with LSTM-based sequence to sequence model. Energy Build 245:111053
42. Himeur Y, Ghanem K, Alsalemi A, Bensaali F, Amira A (2021) Artificial intelligence based anomaly detection of energy consumption in buildings: a review, current trends and new perspectives. Appl Energy 287:116601
43. Chen Y, Wu Q (2011) Design and implementation of PID controller based on FPGA and genetic algorithm. In: Proceedings of 2011 international conference on electronics and optoelectronics, vol 4. IEEE, pp 4–308
44. Garces-Jimenez A, Gomez-Pulido J-M, Gallego-Salvador N, Garcia-Tejedor A-J (2021) Genetic and swarm algorithms for optimizing the control of building HVAC systems using real data: a comparative study. Math 9(18):2181
45. Lauret P, Boyer H, Riviere C, Bastide A (2005) A genetic algorithm applied to the validation of building thermal models. Energy build 37(8):858–866
46. Torquato MF, Fernandes MA (2019) High-performance parallel implementation of genetic algorithm on FPGA. Circuit Syst Signal Process 38(9):4014–4039
47. Elkholy M, Metwally H, Farahat M, Senjyu T, Lotfy ME (2022) Smart centralized energy management system for autonomous microgrid using FPGA. Appl Energy 317:119164
48. Petri I, Li H, Rezgui Y, Chunfeng Y, Yuce B, Jayan B (2014) A modular optimisation model for reducing energy consumption in large scale building facilities. Renew Sustain Energy Rev 38:990–1002. https://doi.org/10.1016/j.rser.2014.07.044
49. Olesen BW, Parsons K (2002) Introduction to thermal comfort standards and to the proposed new version of EN ISO 7730. Energy build 34(6):537–548
50. Zhou Z, Qiu C, Zhang Y (2023) A comparative analysis of linear regression, neural networks and random forest regression for predicting air ozone employing soft sensor models. Sci Rep 13(1):22420
51. Nagel M, Fournarakis M, Amjad RA, Bondarenko Y, Van Baalen M, Blankevoort T (2021) A white paper on neural network quantization. Preprint arXiv:2106.08295
52. Shashank R, Rajagopal M (2025) Machine learning algorithm for optimising comfort cooling in buildings. International J Sci Eng Technol 13(1). https://doi.org/10.61463/ijset.vol.13.issue1.150
53. Ali A, Jayaraman R, Mayyas A, Alaifan B, Azar E (2023) Machine learning as a surrogate to building performance simulation: predicting energy consumption under different operational settings. Energy Build 289:112940. https://doi.org/10.1016/j.enbuild.2023.112940
54. Przybył A (2021) Fixed-point arithmetic unit with a scaling mechanism for FPGA-based embedded systems. Electron 10(10):1164. https://doi.org/10.3390/electronics10101164
55. Constantinides G, Kinsman A, Nicolici N (2011) Numerical data representations for FPGA-based scientific computing. IEEE Design Test Comput 28(4):8–17. https://doi.org/10.1109/MDT.2011.48
56. Ney J, Hammoud B, Dörner S, Herrmann M, Clausius J, Brink S, Wehn N (2022) Efficient FPGA implementation of an ANN-based demapper using cross-layer analysis. Electron 11(7):1138. https://doi.org/10.3390/electronics11071138
57. Dinh TP, Pham-Quoc C, Thinh TN, Nguyen BKD, Kha PC (2023) A flexible and efficient FPGA-based random forest architecture for IoT applications. Internet Thing 22:100813. https://doi.org/10.1016/j.iot.2023.100813

58. Bae Y, Bhattacharya S, Cui B, Lee S, Li Y, Zhang L, Im P, Adetola V, Vrabie D, Leach M, Kuruganti T (2021) Sensor impacts on building and HVAC controls: a critical review for building energy performance. Adv Appl Energy 4:100068. https://doi.org/10.1016/j.adapen.2021.100068
59. Im P, Bae Y, Cui B, Lee S, Bhattacharya S, Adetola V, Vrabie D, Zhang L, Leach M (2020) Literature review for sensor impact evaluation and verification use cases - building controls and fault detection and diagnosis (fdd). Technical report, Oak Ridge National Laboratory (ORNL), Oak Ridge, TN (United States). https://doi.org/10.2172/1649168 . https://www.osti.gov/biblio/1649168
60. Li Z, Huang G (2013) Re-evaluation of building cooling load prediction models for use in humid subtropical area. Energy Build 62:442–449
61. Fanger PO (1970) Thermal comfort. In: Analysis and applications in environmental engineering. Danish Technical Press, Copenhagen, p 244
62. Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Reading, MA, p 412
63. Ouni B (2014) Low power design finite state machine on field programmable gate arrays. Int J Appl Eng Res 9:16341–16352
64. Borowik G (2006) Serial decomposition of finite state machines for FPGA-based implementation. In: Photonics applications in astronomy, communications, industry, and high-energy physics experiments IV, vol 6159. SPIE, pp 927–934
65. Senhadji-Navarro R, Garcia-Vargas I, Guisado JL (2012) Performance evaluation of ram-based implementation of finite state machines in FPGAs. In: 2012 19th IEEE international conference on electronics, circuits, and systems (ICECS 2012). IEEE, pp 225–228

Authors and Affiliations

Mohammed Mshragi[1] · Ioan Petri[1]

✉ Mohammed Mshragi
MshragiM@cardiff.ac.uk

Ioan Petri
petrii@cardiff.ac.uk

[1] School of Engineering, Cardiff University, Cardiff, UK